



Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

2007-12

Requirements to create a persistent, open source, mirror world for military applications

Sanders, Kent L.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/3137>

Downloaded from NPS Archive: Calhoun



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community.

Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**REQUIREMENTS TO CREATE A
PERSISTENT, OPEN SOURCE, MIRROR WORLD
FOR MILITARY APPLICATIONS**

by

Kent Sanders

December 2007

Thesis Co-Advisors:

Don Brutzman

Amela Sadagic

Terry Norbraten

Second Reader:

Approved for public release; distribution unlimited

This Thesis Done in Cooperation with the MOVES Institute

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE
Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 2007	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Requirements to Create a Persistent, Open Source, Mirror World for Military Applications		5. FUNDING NUMBERS
6. AUTHOR Kent Sanders		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.		
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE A
13. ABSTRACT With many free simulations available to developers, it is desirable to marry the existing library and work of these simulations in an attempt to create a completely open source virtual reality environment to facilitate computer aided training and simulation. Key problems associated with developing a large scale Multiple User Virtual Environment (MUVE) are analyzed including appropriate server – client architecture, terrain and object model formats, and overall project scalability. Solutions to these problems are proposed and analyzed, including using existing commercial and open source projects in development, using projects already deployed, or the feasibility of developing a new solution to meet the requirements of this thesis. Advantages, disadvantages, and possible military and educational uses for each of these free simulations and the associated persistent mirror worlds are also analyzed to recommend a direction of action for military and education simulation and training.		
14. SUBJECT TERMS Second Life, There!, X3D, Prim.Blender, Project Darkstar, OLIVE, Serious Games, Mirror World, Google Earth, NASA World Wind, Microsoft Virtual Earth, Persistent World, XML, large-scale virtual environments, simulation, social networking		15. NUMBER OF PAGES 113
16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified
20. LIMITATION OF ABSTRACT UU		

NSN 7540-01-280-5500

 Standard Form 298 (Rev. 2-89)
 Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution unlimited

**REQUIREMENTS TO CREATE A PERSISTENT,
OPEN SOURCE, MIRROR WORLD FOR MILITARY APPLICATIONS**

Kent L. Sanders
Lieutenant, United States Navy
B.S., University of Idaho, 1997

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS AND
SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
December 2007**

Author: Lieutenant Kent L. Sanders, USN

Approved by: Don Brutzman
Thesis Co-Advisor

Amela Sadagic
Thesis Co-Advisor

Terry Norbraten
Second Reader

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With many free simulations available to developers, it is desirable to marry the existing library and work of these simulations in an attempt to create a completely open source virtual reality environment to facilitate computer aided training and simulation. Key problems associated with developing a large scale Multiple User Virtual Environment (MUVE) are analyzed including appropriate server – client architecture, terrain and object model formats, and overall project scalability. Solutions to these problems are proposed and analyzed, including using existing commercial and open source projects in development, using projects already deployed, or the feasibility of developing a new solution to meet the requirements of this thesis. Advantages, disadvantages, and possible military and educational uses for each of these free simulations and the associated persistent mirror worlds are also analyzed to recommend a direction of action for military and education simulation and training.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PROBLEM AND THESIS STATEMENT	1
B.	MOTIVATION	2
C.	SCOPE OF THE THESIS.....	4
D.	THESIS ORGANIZATION.....	5
1.	Chapter II – Problem Overview: MMOG and Serious Games	5
2.	Chapter III – Related Work.....	6
3.	Chapter IV – 3D Graphics Interoperability.....	6
4.	Chapter V – Networking Interoperability	6
5.	Chapter VI – Informal Comparison of Virtual Environment Features.....	6
6.	Chapter VII – Conclusions and Recommendations.....	7
II.	PROBLEM OVERVIEW: MMOG AND SERIOUS GAMES.....	9
A.	INTRODUCTION.....	9
B.	SERIOUS GAMES AND MMOGS.....	9
1.	Introduction to Serious Games	9
2.	Motivation.....	12
3.	Changing Culture	15
4.	Useful Learning.....	16
5.	Skills Learned From Games	18
6.	Game Development.....	24
a.	<i>New Opportunities</i>	25
b.	<i>Reusability</i>	25
7.	Bad Publicity	25
C.	CONCLUSION	26
III.	RELATED WORK	27
A.	INTRODUCTION.....	27
B.	ARMY GAME PROJECT	27
C.	METAVERSE ROADMAP	29
1.	Background and Introduction	29
2.	Bases for Speculation.....	30
3.	Enabling Trends.....	31
4.	Lifelogging	34
D.	MULTIVERSE AND SOCIAL NETWORKING.....	34
E.	SAVAGE X3D WORLD MODELS	35
F.	SECOND LIFE FOR RESEARCH.....	38
G.	CONCLUSION	38
IV.	3D GRAPHICS INTEROPERABILITY	39
A.	INTRODUCTION.....	39
B.	MOTIVATION	39

C.	3D MODELS	40
1.	<i>OLIVE.....</i>	<i>40</i>
2.	<i>There!</i>	<i>40</i>
3.	<i>Second Life</i>	<i>41</i>
a.	<i>Prim. Blender</i>	<i>42</i>
b.	<i>OBJ Importing Script</i>	<i>45</i>
c.	<i>Maya / 3DS Max importing.....</i>	<i>45</i>
4.	<i>Project Darkstar.....</i>	<i>45</i>
D.	OPERATING REQUIREMENTS	46
1.	<i>OLIVE.....</i>	<i>46</i>
2.	<i>There!</i>	<i>46</i>
3.	<i>Second Life</i>	<i>47</i>
4.	<i>Project Darkstar.....</i>	<i>48</i>
E.	CONCLUSION	48
V.	NETWORKING INTEROPERABILITY	49
A.	INTRODUCTION.....	49
B.	MOTIVATION	49
C.	OLIVE.....	50
D.	THERE!	51
E.	SECOND LIFE.....	52
F.	PROJECT DARKSTAR	53
G.	CONCLUSION	53
VI.	INFORMAL COMPARISON OF VIRTUAL ENVIRONMENT FEATURES ..	55
A.	INTRODUCTION.....	55
B.	OLIVE.....	55
1.	Advantages.....	55
a.	<i>Visual Appeal</i>	<i>55</i>
b.	<i>Military Collaboration</i>	<i>56</i>
c.	<i>Voice Communication</i>	<i>57</i>
d.	<i>Commercial Development.....</i>	<i>58</i>
e.	<i>DIS / HLA Gateway Connections.....</i>	<i>59</i>
f.	<i>Vehicles.....</i>	<i>60</i>
2.	Disadvantages.....	60
a.	<i>Commercial Development.....</i>	<i>60</i>
b.	<i>Model Format.....</i>	<i>61</i>
c.	<i>Terrain Format</i>	<i>61</i>
3.	Recommendations and Uses.....	61
C.	THERE!	62
1.	Advantages.....	62
a.	<i>Population</i>	<i>62</i>
b.	<i>Recordable.....</i>	<i>63</i>
c.	<i>World Creation.....</i>	<i>63</i>
d.	<i>Weather Control.....</i>	<i>63</i>
e.	<i>3DS Max Support.....</i>	<i>63</i>
f.	<i>Voice Communications.....</i>	<i>64</i>

g.	<i>Apearance</i>	64
2.	Disadvantages	65
a.	<i>Limited Scripting</i>	65
b.	<i>Model Submission</i>	65
c.	<i>Commercial</i>	65
3.	Recommendations and Uses	65
D.	SECOND LIFE	66
1.	Advantages	66
a.	<i>Population</i>	66
b.	<i>Open Source</i>	66
c.	<i>Requirements</i>	66
d.	<i>Hosted</i>	67
e.	<i>Web Integration</i>	67
f.	<i>Movement</i>	67
g.	<i>Tutorials</i>	68
h.	<i>Avatar Customization</i>	68
2.	Disadvantages	69
a.	<i>Graphically Unappealing</i>	69
b.	<i>Chaotic Content</i>	70
c.	<i>Voice</i>	70
d.	<i>Security</i>	70
e.	<i>Terrain</i>	71
f.	<i>No Playback</i>	71
g.	<i>Proprietary</i>	71
h.	<i>Cost</i>	71
3.	Recommendations and Uses	72
a.	<i>Language Training</i>	72
b.	<i>Distance Education</i>	72
c.	<i>Recruiting</i>	73
d.	<i>Teleconferencing / Virtual Collaboration</i>	73
E.	PROJECT DARKSTAR	73
1.	Advantages	73
a.	<i>Open Source Java</i>	73
b.	<i>Developing User Community</i>	74
c.	<i>Configurability</i>	75
d.	<i>Cost</i>	75
e.	<i>Terrain</i>	75
f.	<i>Plug-in Design</i>	76
g.	<i>Scalability</i>	76
h.	<i>Persistence</i>	77
i.	<i>Fault Tolerance</i>	77
2.	Disadvantages	77
a.	<i>Work In Progress</i>	77
b.	<i>No User Scripting</i>	77
c.	<i>Limited Documentation</i>	78

3.	Recommendations and Uses.....	78
a.	<i>Persistent Mirror World Simulation.....</i>	78
F.	CONCLUSION	78
VII.	CONCLUSIONS AND RECOMMENDATIONS.....	79
A.	CONCLUSIONS	79
B.	RECOMMENDATIONS / FOLLOW ON WORK	81
1.	Develop a Fully Persistent Mirror World Simulator.....	81
a.	<i>Where to Start?.....</i>	81
b.	<i>Networking</i>	82
c.	<i>Environment.....</i>	83
2.	Recommended Uses	85
a.	<i>War gaming</i>	85
b.	<i>Distance Learning.....</i>	85
c.	<i>Recruitment.....</i>	86
d.	<i>Social Networking</i>	86
e.	<i>Information</i>	87
f.	<i>Military Officer Training.....</i>	88
3.	Establish a Second Life Recruiting and Campus Presence.....	88
a.	<i>Campus Presence</i>	88
b.	<i>Recruiting.....</i>	89
c.	<i>Public Relations</i>	89
LIST OF REFERENCES		91
INITIAL DISTRIBUTION LIST		93

LIST OF FIGURES

Figure 1.	An illustration of the variety of serious game objectives that is possible [From USC Gamepipe Laboratory]	10
Figure 2.	One of many thousands of developed flight simulators in use [From University of Bristol].	14
Figure 3.	There is a large overlap in today's computer games and the area of simulations.....	16
Figure 4.	The curve shows the increase in performance in a task over trials.....	20
Figure 5.	Difficult visual target differentiation can be improved through computer gaming in America's Army.	21
Figure 6.	Computer gaming can help in assessing target threat prioritization, as in America's Army.....	22
Figure 7.	A soldier confronts multiple targets in the field in America's Army.	23
Figure 8.	A user of America's Army is involved in urban operations.....	28
Figure 9.	Moore's Law depicts an exponential increase in processing resources [From Metaverse Roadmap].	31
Figure 10.	Kryder's Law predicts exponential growth in computer storage capacity over time [From Metaverse Roadmap].	32
Figure 11.	Exponential growth in most areas of computing resources is predicted [From Metaverse Roadmap].	33
Figure 12.	Amateur .X3D model is displayed using Media Machines Flux Player.....	37
Figure 13.	The Prim.Blender second build in Mandriva Linux. 2.16 kernel is used to create basic geometry.....	43
Figure 14.	The Prim.Blender version 0.4.0 on Blender 2.42a (running on Windows XP OS) shows the basic primitive object types.....	44
Figure 15.	There are six key technologies that drive computer games [From <i>Serious Games: Games That Educate, Train, and Inform</i>]]......	50
Figure 16.	Simulations have progressed from professional virtual trainers to ubiquitous web applications for all users [From Metaverse Roadmap].	51
Figure 17.	The creation of an OLIVE soldier avatar is achieved through sliders.....	56
Figure 18.	The OLIVE avatar mans a fully configurable checkpoint.....	57
Figure 19.	This figure depicts OLIVE's agile game development concept [From Forterra].	58
Figure 20.	An OLIVE soldier pats down a citizen for hidden weapons at a manned checkpoint.....	59
Figure 21.	One of the strengths of OLIVE is its use of vehicles.....	60
Figure 22.	There are many options for avatar creation in Makena's There!.....	62
Figure 23.	There! has a uniform world appearance.....	64
Figure 24.	The De-babbler script is translating local text into another language from within Second Life.....	68
Figure 25.	Second Life has the most configurable avatar of the popular simulations.	69
Figure 26.	MPK20 provides a great example of Project Darkstar's capabilities [From Sun System's <i>Project Darkstar</i>].	74

Figure 27. MPK20 is a virtual building extending upon IBM's Menlo Park facility
using Project Darkstar.....76

LIST OF TABLES

Table 1. Key features that are available in each simulation are denoted by an “X”80

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AI	Artificial Intelligence
API	Application Programming Interface
ASF	Acceleration Studies Foundation
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CLEP	College Level Examination Program
DEM	Digital Elevation Map
DIS	Distributed Interactive Simulation
DOD	Department of Defense
GMT	General Military Training or Greenwich Mean Time
GUI	Graphical User Interface
HLA	High Level Architecture
HUD	Heads-Up Display
IDC	International Data Corporation
KML	Keyhole Markup Language
LSL	Linden Scripting Language
IED	Improvised Explosive Device
ICS	Internet Connection Sharing
MMOG	Massively Multiplayer Online Game
MOVES	Modeling, Virtual Environments, and Simulation
MUVE	Multiple User Virtual Environment
NAT	Network Address Translation
NPS	Naval Postgraduate School
OLIVE	On-Line Interactive Virtual Environment
OR	Operations Research
PC	Personal Computer
SAVAGE	Scene Authoring and Visualization for Advanced Graphical Environments
SDK	Software Developer's Kit
SL	Second Life
SMAL	Savage Modeling and Analysis Language
TEEM	Teachers Evaluating Educational Multimedia
UAV	Unmanned Aerial Vehicle
VOIP	Voice over Internet Protocol
WWW	World Wide Web
X3D	Extensible 3D
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my wife, Gwendolyne, for her years of support, love and patience. Without her I would not be the person I am today, and I would not have finished this thesis.

I would like to thank the Modeling, Virtual Environment, and Simulation (MOVES) faculty at the Naval Postgraduate School (NPS) for providing an incredible learning environment. I would like to individually recognize Don Brutzman, Amela Sadagic, Terry Norbraten, and CDR Joseph Sullivan for their help and guidance through a thesis with a timeline filled with extraordinary circumstances. Finally, I would like to thank the staff at NAVSOC, especially Jim Trump and CAPT Diane Cangelosi, for their understanding and patience while my attention was pulled in two different directions.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM AND THESIS STATEMENT

Developing virtual environments is a task of enormous complexity and resources. Often this cost and complexity have made developing them prohibitive for the United States military, civilian businesses, and educational institutions; or the development of these virtual environments and simulators required the contracting of large software development teams at the cost of many millions of dollars and man-hours. However, freely developed virtual environments, simulators, and social networks have recently become available that are poised to redefine the World Wide Web (www), as well as the creation, purchase and utilization of virtual environments and simulations. The possibility of creating a Metaverse of networked three-dimensional (3D) web applications built around, and fully interactive with, the two-dimensional (2D) internet appears to be dawning on the horizon.

These free online virtual environments, virtual and mirror worlds, and social networks provide a unique opportunity as well as new difficulties in harnessing their effectiveness for military or educational purposes. This thesis attempts to outline a future path that can be taken to overcome some of these hurdles in establishing a presence in the 3D web and Metaverse. It examines what is required to use the existing military model database of the Scene Authoring and Visualization for Advanced Graphical Environments (SAVAGE) project inside these free virtual environments. It also analyzes how the structure of the Extensible 3D language (X3D) makes it an excellent solution as the open source standard for 3D modeling in the World Wide Web as well as future modeling engines designed for military applications. It examines the feasibility of creating an open source mirror world using programs such as X3D Earth or Google Earth, specifically for military training and simulation that is fully persistent and user-content created. Specific applications that the military can utilize are analyzed to determine which technology might best serve their needs and in what capacity to provide a future roadmap of development of this persistent mirror world.

B. MOTIVATION

Rather than creating a virtual world for its own purposes, the military, as well as most organizations, needs to embrace the power of collaboratively developed, user-created worlds, and thus harness the freely available libraries of content developed by other users for other applications. As wasteful of resources and under utilizing a powerful tool as developing a strictly military version of the Internet might be, so too is continuing to develop standalone simulations that do not interact and network without leveraging existing open source projects. Program managers are beginning to understand this realization as more and more contracted projects are required to have their source code be open in order to prevent locking the government into one software vendor. Thus GNU and other open-source projects are becoming more prevalent in the software development realm.¹ By drawing on the vast resources of the content of Second Life, There!, OLIVE (On-Line Interactive Virtual Environment), OpenSim, Project Wonderland, Moove, and other developing, free virtual environments and social networks, the military might establish a very cost effective means of teleconferencing, simulations, training, information repositories, virtual chat, networking, and all of the other uses that are being developed for these free environments, but used within its own persistent mirror world where they are free to control the content, delivery, and security of these applications. However, the unique operating requirements of the military will require the creation of a mirror world with different characteristics than any single source, commercially developed application that is being developed to date. This necessitates the use of an open standard for portability and extensibility of 2D and 3D objects and applications, developed in other worlds of the emerging Metaverse.

It is far more cost effective to start out with a completely fleshed out virtual environment, filled with everything a user can imagine, and then eliminate undesirable content until an acceptable virtual world remains as opposed to creating that virtual world from scratch. Eliminating content only requires a keystroke; building content requires

¹ Government Computer News, GNC, "Navy CIO approves open-source software use," http://www.gcn.com/print/26_14/44462-1.html?topic=defense-technology&CMP=OTC-RSS, last accessed August 2007.

significant efforts and resources. This is the power and versatility of using available resources to match current training requirements. Harnessing the open source and open standards library of X3D with the open source virtual environments provides a “pick and choose” method of assembling free virtual environments and simulations and a mirror world to meet those training requirements. Developing a mirror world (a virtual environment based up upon very accurate geometry of the real world) using the lessons learned, networking and architecture from the Massively Multiplayer Online Gaming (MMOG) community can be a cost affordable way to implement this new virtual trainer and simulator. Although no single source exists that entirely meets the needs of a military and educational simulation, being able to choose the best pieces and parts from these simulations to flesh out a new simulator will drastically lower the development cost and expenditure. This will remove the development of the training environment done only by large software development companies isolated from the military component they are designing for, and will allow for development by in house design teams that are much closer to the required design specifications and are subject matter experts themselves. This moves the training environment and simulation from on shore training facilities and warehoused computers to forward deployed warriors using commodity-based personal computers (PC). Currently there is no ability for a small, forward deployed command to effectively simulate an upcoming event using its own resources, and these are often the users who need this ability the most!

This transformative effect of placing training and simulation at the fingertips of the war fighters, where it can most effectively be employed, can have other very useful military applications that are examined later in this thesis. The majority of deployed commands and units, which are doing some of the most important work of all military branches, have *no* simulation tools to analyze for situations they might encounter in a short term scenario. They must rely on command experience or a simulation result run from a separate facility to help formulate a solution to the problem. As discussed later, combat operations that rely solely on experience is an option that has proved lethal and ineffective, as in the case of pilots since World War II. Relying on a simulation solution returned from another facility also runs the risk of error and issues of timeliness. What is

needed is a small, fast, easy to use training and simulation suite that is able to run effectively on commodity based personal computer equipment. This simulation suite would preferably be open sourced, extensible, fully networked, and based on open standards for future development. The generated terrain providing the background for this simulation would draw upon the power of other software solutions such as X3D Earth, Google Earth, NASA World Wind, and Microsoft Virtual Earth to provide a fully interactive scenario based upon real world terrain for any portion of the Earth in a fully persistent mirror world. Ideally, it would be able to pull from *all* of those sources as necessary, with the power and flexibility of either an open source standard of file interchange or a software loader or conversion program. When necessary terrain features need to be updated, enemy or friendly combatant units added, or removed from the scenario, or plug-ins to the simulation engine are required to create a more accurate simulation, the changes could be made by personnel who need not be proficient programmers or modelers. This is part of the powerful basis of X3D – creatable 3D content that can be created and modified by anyone, not just professional software designers. Since the X3D standard is written in Extensible Markup Language (XML) as opposed to a proprietary modeling format, the model itself contains its own information in its metadata to be used in the simulation. This greatly facilitates version control, automation, modifiability, and extensibility, even for remote units with poor network connectivity. This should be the future goal of usability in military simulation and training software: to be usable with the current knowledge and training level of the war fighter, not the programmer, and to be forward deployed at the fingertips of these war fighters.

C. SCOPE OF THE THESIS

This thesis explores the following topics:

- It explores the feasibility of developing a software program capable of converting existing models in the X3D format into commercially developed virtual worlds, such as There!, Second Life, and Project Darkstar.

- It examines the type of networking architecture each of these environments uses, as well as the architecture used by MMOG, to help determine how this mirror world can best set itself up to take advantage of these other development efforts.
- It analyzes how the world and models are created, and the rules that are defined for each virtual world to determine if it is well suited for importing into a military mirror world.
- It examines existing free virtual environments and simulations to determine the advantages and disadvantages, and possible uses, each one possesses for military and education applications to determine how it can be connected in a simulation Metaverse.
- It determines how best to proceed to move tools of simulation and online virtual training forward to the deployed commands, as well as shore-based installations.
- Lastly, it provides a roadmap forward for educational and military application of an open source mirror world in the development and creation of a new solution.

D. THESIS ORGANIZATION

The remainder of this thesis is organized into six chapters:

1. Chapter II – Problem Overview: MMOG and Serious Games

Chapter II describes the expanding field of serious gaming, and the explosion of massively multiplayer online games. How can the military best utilize these new tools? What are the requirements inside each virtual world for graphics, networking, and user interaction to be useful for training? What lessons have game designers learned from gaming and MMOGs that can be useful when constructing a virtual environment capable of supporting thousands of educators, trainers, users, students, and military members?

2. Chapter III – Related Work

This chapter analyzes what previous work has been done in the area of X3D model translation, virtual world creation, and development of social networks, trainers, and simulators that are capable of meeting the requirements of this thesis.

3. Chapter IV – 3D Graphics Interoperability

By analyzing how the existing virtual worlds develop in-game models, software designers can determine if offline loaders and modeling programs can be used to populate these worlds, and whether or not the virtual environment can be connected in Metaverse fashion. This chapter also analyzes the possibility of utilizing existing military models in the X3D format for these virtual environments to provide a roadmap of connectivity between the environments and the X3D standard.

4. Chapter V – Networking Interoperability

This chapter examines the protocols used between these virtual environments, such as There! and Second Life, as well as other developing technologies such as OpenSim, Project Darkstar, etc. By analyzing these network protocols, can tools such as X3D better be designed to operate inside these environments? Can these virtual environments be seamlessly integrated or are they proprietary? What options are available to serve as the backdrop for our virtual environment? Is it possible to develop a virtual environment simulation that is using a digital projection of the real world as its geography?

5. Chapter VI – Informal Comparison of Virtual Environment Features

This chapter performs an informal comparison of the most relevant virtual environments and simulations to answer the questions posed by this thesis. As each virtual environment continues to evolve and improve over time, only general aspects of each virtual environment are analyzed.

6. Chapter VII – Conclusions and Recommendations

Chapter VII attempts to predict the future developments and research that can follow from this thesis. It also details follow-on work that can further improve the groundwork identified by this research as well as recommendations based upon the knowledge and lessons learned from this study. This provides a basis for the development of a fully persistent simulation with the geometry of the real world as the virtual environment.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM OVERVIEW: MMOG AND SERIOUS GAMES

A. INTRODUCTION

Can video games be developed where the primary emphasis is placed on learning and education, and fun has been placed in a secondary role? Can video game software that has already been used to educate be redesigned or redeployed to become a tool of training and simulation? By analyzing the evolution of computer technology in simulations that have successfully transitioned to accepted training platforms, such as the flight simulator, game designers can learn why serious games hold an attractive position in the future of education. Developers can also learn key insights in a particular genre of gaming, the MMOG, that provide valuable insight into the networking, design, and development of a persistent world based upon real world geography. By using these lessons learned it becomes possible to create a project that parallels the MMOG game design architecture, but with the emphasis placed on training and usability of the engine within a real world environment.

B. SERIOUS GAMES AND MMOGS

1. Introduction to Serious Games

Serious games might sound like an oxymoron, something being both fun and serious at the same time, but a “fun game” and a “serious game” are not mutually exclusive. A serious game, then, can be defined as *a game in which education (in all its varying forms) is the primary goal of the game, rather than entertainment*. In *A Theory of Fun*, Koster defines “fun” as a side effect of learning something new, something that students “get”. In essence, fun is essentially a positive feedback mechanism for successfully absorbing a pattern to get participants to repeat an activity that the he is experiencing over and over. Therefore, if a player does not find a game fun, he is unlikely to choose to play it again. If a player has beaten the game, or progressed to a point where the game has become repetitive, he will also probably become bored and quit

the game. Thus, according to Koster, every game qualifies as “edutainment.” This might seem dubious to some, who see entertainment and education as opposite endpoints on a spectrum of possibilities. However, not only are education and entertainment *not* in conflict, but there are many places where the two not only overlap, but are themselves better defined from each other. Koster goes on to say that fun is not only *not* frivolous, but fundamental to human nature and required for survival. This means game designers are in a unique position to help educators create products that motivate and engage the participant, and educators are in a unique position to help game developers create products that train and educate more effectively. Figure 1 depicts the many avenues current serious game developers are exploring.

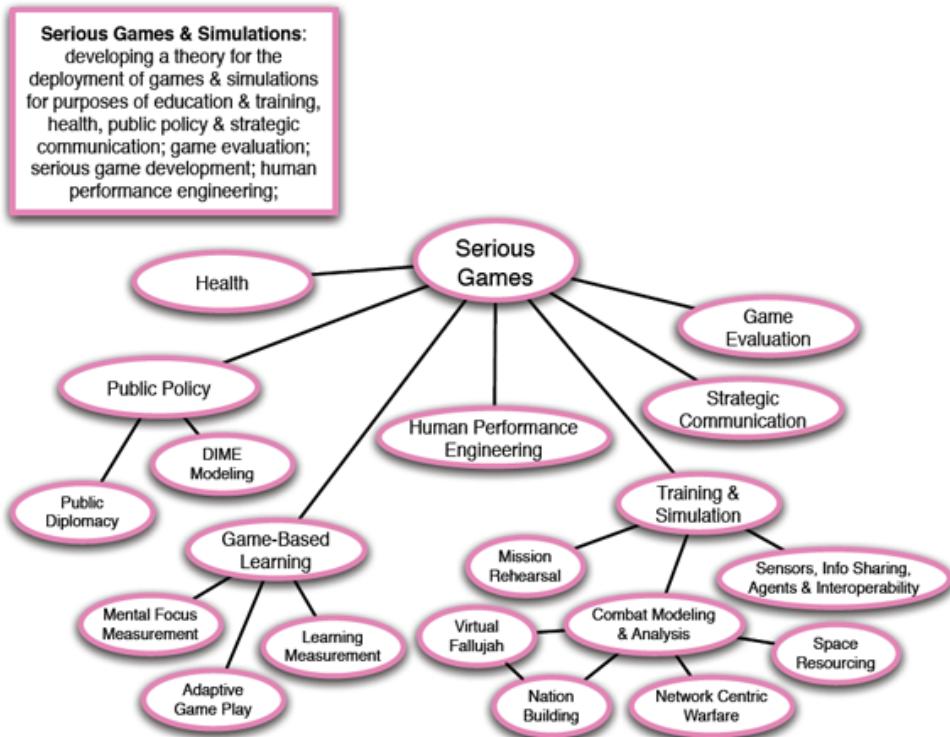


Figure 1. An illustration of the variety of serious game objectives that is possible [From USC Gamepipe Laboratory]

Games are a distillation of cognitive schema. Gamers try to maximize their results within the game: see the game pattern, figure the optimal path, then apply it and re-apply it. If cheating is possible, gamers will often take it, in effect applying an “Alexandrine solution to a Gordian knot” problem. In Greek mythology, an oracle revealed that the knot tied by King Gordius of Phrygia could only be unraveled by the future conqueror of Asia. According to tradition, in 334 BC Alexander the Great, unable to untie the knot, cut it in two with his sword². Although lateral thinking like this is considered cheating in most games, it is precisely the type of problem solving skills imparted in education: applying non-lateral decision making to routine, real-world problems to maximize a desired end result. Although this places entertainment game designers in a hopeless task of creating content that can’t be bypassed or “gamed” (some cheating practices are also called “gaming the system”) by the player. The lack of variation or some systemic error in the game when it is possible to “game the system” is also practiced in Serious Games.

Game designers, especially MMOG game designers, typically shift the end point of the game in a moving target fashion with each new expansion. This moving target is the “treadmill” of MMOGs. How slowly can game developers keep the game player from advancing through content, thus requiring the developers to recreate new content, but not frustrate the player? This treadmill must be slow enough to allow for ever increasing model and art development, but still show sufficient game progress for the players to not quit the game. The answer that the MMOGs have found is using increasingly proficient artificial intelligence, and an even better, longer lasting solution: playing against another human opponent. This causes the game to vary with every play session and provide the user with the fun experience in a new way every session. Most long-lasting games have been competitive because they lead to an endless supply of similar yet subtly varied puzzles that force us to think through the problem every time, while still “getting” the game. Most adults (and children too) prefer to play chess as opposed to tic-tac-toe for this very reason. A well-designed simulation or social network

² The Free Dictionary by Farlex, “Alexandrian solution”
<http://encyclopedia.farlex.com/Alexandrian+solution>, last accessed December 2007.

has to rely on both of these techniques to engage and involve the participant, as well as keep them coming back for more and more play sessions.

2. Motivation

Prensky argues in “Digital Natives, Digital Immigrants” that the “*sine qua non* of successful learning is motivation: a motivated learner can’t be stopped.” It can be argued that today’s learning audience, from adolescents to adults, civilian to military, and business to education is not directly motivated by the content and method teachers are using to engage them. However, this “deficiency” of the education system to effectively engage and motivate the student is exactly the strength and expertise of the \$30 billion worldwide computer and video gaming industry.

Games have the power to teach, train, educate, and inform. They have the power to bring people of diverse backgrounds together in a social environment and to break down social barriers. Video games are maturing as a medium and tool, and with that maturity comes an opportunity and responsibility to do more than just provide entertainment for the masses. Children are taught games as a social activity to help prepare them for organized learning and social interaction. Games are being developed and created to teach modern cognitive schema such as how to effectively tackle global warming and how to minimize risk to cancer³. For a list of some of the most impressive edutainment games currently being developed for every area of use from politics to business to the military, visit socialimpactgames.com.

Games can also reveal more about a person’s character in a short period of time than almost any other means of contact. In *The Republic*, Plato emphasized the connection between play and education by recommending games for the education of children. He saw philosophical discourse – wordplay – as an educational game to be used throughout the rest of our lives. He said, “You can discover more about a person in an hour of play than in a year of conversation.” Military officers have been using games

³ Social Impact Games, “Cancer Re-Mission,” <http://www.socialimpactgames.com/modules.php?op=modload&name=News&file=article&sid=39&mode=thread&order=1&thold=0>, last accessed December 2007.

like Chess and Go for thousands of years to teach important concepts of strategy and tactics. With the evolution of warfare changing the focus of military operations from the destruction of the enemy forces to the destruction of the enemy's will to fight, more defined and nuanced thinking is required by military officers. Often this training is best learned through war gaming.

After World War II, the US military's use of war games was largely limited to the field of Operations Research (OR) at the strategic level of warfare. This was due to limitations of the computing power of the time, and the type of non-fighting warfare unique to the Cold War, which was primarily fought as a battle of resources. With the rise in computing power, other games and simulations that have become incredibly effective military tools have surfaced. For example, the vehicle training simulator has become a ubiquitous staple of every branch of military training, especially flight simulators.

Though flight simulation training today is a mandated requirement for both civilian and military pilots, it took over 20 years before the simulators were accepted as a viable training technique. Many observations and experimental results are still being conducted to truly discover their power and effectiveness, and to also determine the best means to utilize simulation training. Some results are already plainly evident: as pilot experience with his aircraft in WWII grew, the death rate of pilots decreased. As an individual pilot flew more and more missions, the likelihood of his death decreased. Thus, for a new pilot, the first run was the most dangerous, and it was determined that pilots were learning to survive through experience. The powerful tool of simulation provided the only means to allow pilots to learn these survival skills without the risk of death. Today, all flight training involves hundreds of hours in simulators, as in Figure 2, before a pilot is allowed to take the controls of a real aircraft.



Figure 2. One of many thousands of developed flight simulators in use [From University of Bristol].

The improvement of the flight simulator is being rediscovered with the advent of more powerful computer networking abilities. Now it has become possible for trainees to be operating in the same battle space, as both friendly and enemy combatants. These vehicle simulators have progressed beyond procedural training to tactical training, providing the real world combat experience necessary to ensure pilot survival. This is a powerful new development with consequences and capabilities that are still being analyzed. Furthermore, simulations are no longer tied to running one “type” of scenario: pilots are no longer only running flight simulations, but are capable of interacting with networked ground units that are also running simulations. This giant leap in simulation training is expected to continue to grow exponentially in usage, capabilities, and other factors, as is examined in Chapter III.

Games and game systems today have evolved in the past 30 years to become incredibly effective tools at player engagement. According to Prensky in *Digital Natives*,

Digital Immigrants: “the ability to keep people in their seats for hour after hour, day after day, at rapt attention, actively trying to reach new goals, shouting with glee at their successes, determined to overcome their failures, all the while begging for more.” This attitude toward gaming and computer technology is often the completely opposite attitude these same students display towards school and education in general. Yet it is the exact disposition (interested, competitive, goal-oriented, dedicated, cooperative and social) teachers are attempting to create within students that is so well elicited by the gaming culture. It makes great sense to attempt to merge the motivation of gaming with the content of learning to create more effective and engaging learning environments.

Games also have the power to heal. Games that require simple puzzles to be solved like Concentration or Mahjongg are used as therapy for patients suffering from learning disorders and brain trauma. These games can help to learn (or relearn) skills like memory retention, pattern recognition, category sorting, and motor skills. BreakAway games, a leader in the serious gaming industry, created a diving simulation game called Crate for the United States Navy which uses a digital elevation map (DEM) and multi-beam sonar data to generate a geospatially accurate, highly detailed visual simulation of the 3D aquatic environment. It has also been used as entertainment to help relieve pain in children undergoing chemotherapy. Although many uses have been discovered for serious gaming, there are many, many more that have yet to be developed and implemented. However, there is no doubt that serious gaming has a future in education, health care, training, and many other fields of study.

3. Changing Culture

According to “Digital Natives, Digital Immigrants,” The amount of time today’s young people spend playing computer and video games, estimated at 10,000 hours by the time they are 21, and often in multi-hour bursts, belies the “short attention span” criticism of educators. The gaming culture demographic has also been slowly changing as well. Years ago the core gaming constituent was adolescent young boys, while today it is increasingly girls and children of all ages, races, economic classes, and social groups. In the America of today, one would be hard pressed to find a child who doesn’t play

computer or video games of some form. Children often display a much higher computer and video game proficiency than parents. The types of games that these players are playing have evolved as well. As in Figure 3, today's most popular computer games can also be classified as virtual environments and simulations. There can be no doubt that video games are a part of our culture as much as books, movies, the internet, radio, and television are. Yet gaming itself can have a negative public image since it is the newest form of information and entertainment media on that list, despite its possibility of becoming one of the most powerful tools to educate and train.

Not only have the digital natives become better at spreading their attention over a spectrum of activities and inputs, but they are also more capable of multitasking and parallel processing, taking in information and making accurate and effective “twitch” decisions. These are distinct advantages to military members achieved through game play, and these are not the only advantages.

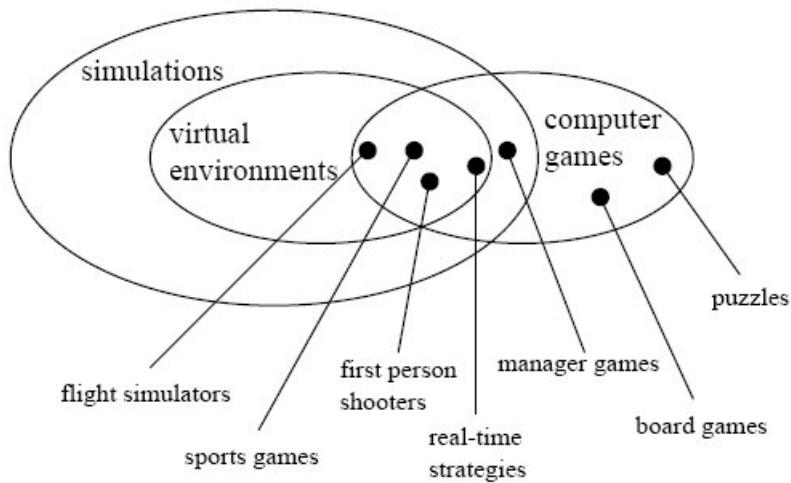


Figure 3. There is a large overlap in today's computer games and the area of simulations.

4. Useful Learning

Not only do game players learn to do task-oriented goals such as flying airplanes, driving cars, develop cities and civilizations, balance and manage systems of dependent variables and resources, and play sports; they are also learning information and skills on a

deeper level. A UK study conducted by Teachers Evaluating Educational Multimedia (TEEM) has shown that some commercial video games can help children learn logical thinking and computer literacy at much greater rates than traditional education techniques.⁴ Students learn to take information in from many different inputs and sources, process it, and decide and act on that information quickly and efficiently. They learn to deduce the game's rules, operating procedures, and goals in order to compete effectively and efficiently against increasingly complex artificial intelligence programs as well as human opponents who are also attempting to make the same deductions and inferences. They learn to understand complex systems of variables through experimentation, devote hours to advancing through "levels" of increasing complexity and difficulty, and learn to collaborate and work with others to achieve goals unobtainable to one individual. Many skills can be taught only by performing an action, and many lessons are best taught through the mechanism of failure. However, some vital lessons are too costly or dangerous to allow failing in the real world. These situations are perfect candidates to be taught virtually. For example, allowing soldiers to enter complex 3D simulations with real-world physics models in a non-lethal simulation can help them perform in a real world lethal environment, in effect giving them the advantage of the veteran combat experience that has proven useful in keeping pilots alive since World War II.

Many adults might be unaware that games are no longer the single-player interaction between a player and the game console that was the norm for games developed in their childhood. Today's most influential and best selling games are Massively Multiplayer, with games like World of Warcraft boasting over 8 million subscribers with hundreds of thousands of players playing concurrently. Everquest still boasts of player base of around one million players 8 years after its launch. Many of these MMOGs are no longer strictly "games" at all, but elaborately interwoven simulation and socialization programs boasting hundreds of thousand of users, such as There! and Second Life. These MMOG players often form elaborate gaming

⁴ Teachers Evaluating Educational Multimedia, "Report on the Educational Use of Games," http://www.teem.org.uk/publications/teem_gamesined_full.pdf, last accessed August 2007.

organizations, such as guilds, clans, and leagues, to be better able to advance in the game and more effectively compete with and against each other. Many games have no definable victory conditions, but instead are set as environments where gamers can continue to play for countless hours a day, for duration of several years.

Games are also capable mechanisms for conveying a point of view, sometimes unintentionally or only as a consequence of the storyline. However, there is no reason why a game can't be designed with the express intent of conveying a certain viewpoint or position.

Prensky argues that today's children that have been exposed to such a gaming environment are able to do and understand much more complex tasks, such as reasoning through complex systems, driving and flying accurately modeled vehicles with increasingly complex game physics, and navigating hundreds if not thousands of gaming menus and interfaces. Children today are likely to spend hundreds of hours driving and flying very detailed computer simulations long before being legally allowed to drive a vehicle. As a result of an outmoded school curriculum, children in school often feel as if the curriculum is a depressant, and this feeling only gets worse as the child develops faster than the school curriculum advances. The "digital immigrant" teacher often knows little of the environment their pupils are experiencing, from their online gaming to their online sharing, exchanging, programming, and socializing. This divide worsens as the child grows older, making it impossible for the teachers to design their learning in the same language and at the same rate the students are used to. This leads to the student feeling the subjects they are studying are outdated and irrelevant. Prensky goes so far as to state that "young people of today understand instinctively that their games are their very best teachers." Embracing this gaming culture and harnessing it as a tool to educate those children, rather than attempting to compete with their attention to gaming, can yield powerful results.

5. Skills Learned From Games

Video games already effectively teach hand-eye coordination and other motor skills, 2D and 3D spatial relationships, 2D and 3D spatial geometry and shapes, and an

“educational curiosity” by rewarding players that seek out new information in unexpected places. In *What Video Games Have to Teach Us about Learning and Literacy*, James Paul Gee writes:

Many good computer and video games are long, complex, and difficult, especially for beginners. People are not always eager to do difficult things. Faced with the challenge of getting them to do so, two choices are often available. Teachers can force them, which is the solution schools use. Or, a temptation when profit is at stake, software developers can dumb down the product. Neither option is open to the simulation industry, at least for the moment. They can’t force people to play, and most avid players don’t want their games dumbed down.

So how do game developers get new and inexperienced players to invest the time it takes to learn their new simulation with every new release? This is a difficult question for which there is no easy answer. However, progress has been made in the form of in-game trainers and tutorials that teach selected portions of the game in incremental steps of difficulty. These games and trainers make one new tool or concept available to the user in each level or round of game play to teach the fundamentals, but not overwhelm the player. This allows immediate immersion, game play, and satisfaction with the product while still educating and informing the player of the rules of the system. This effectively allows user to play, learn and perform before they have become technically competent, which is a very effective means of learning unique to simulation. According to Stitt in *Games That Make Leaders: Top Researchers on the Rise of Play in Business and Education*, this allows users to solve problems that they are not yet good at, gain immediate feedback on each solution attempt, and try again if unsuccessful.

There are numerous other skills learned and developed by online video gaming of use to the military:

Improved ability to multitask – Any watch station in the military is often a low level of baseline activity during the normal stretches of the watch, followed by a very frenzied and hectic pace of activity when a casualty, contact, or other significant event occurs. Furthermore, the initial learning curve of becoming proficient on a watch station

or job is lessened by the ability to effectively multitask information until proficiency at that job or watch station is attained, as is demonstrated in Figure 4.

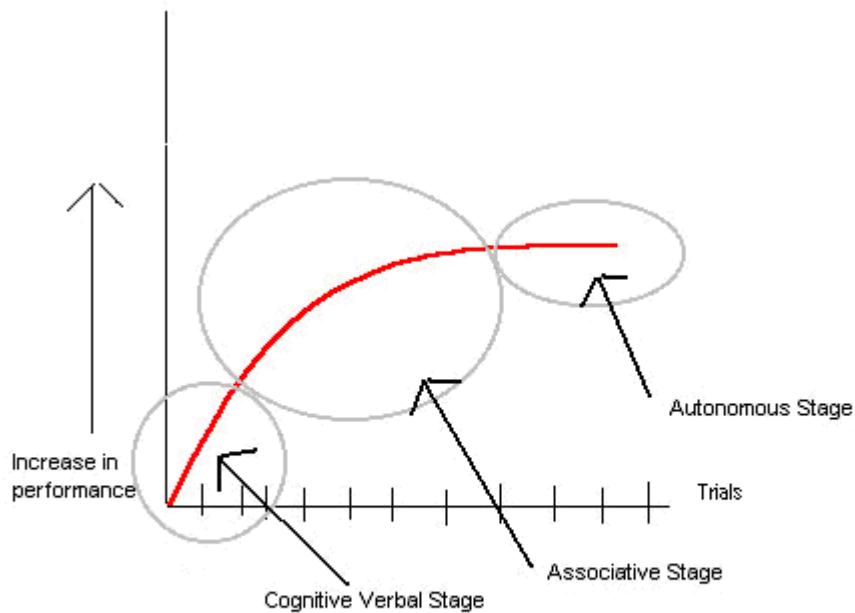


Figure 4. The curve shows the increase in performance in a task over trials.

Improved target differentiation – Often called “twitch” gaming, user response to target differentiation, “attentional blink” (how easy it is to capture someone’s attention), and other basic visual skills are vastly superior in gamers compared to their non gaming counterparts.⁵ In Figure 5, the gamer must recognize the very small target in the field (look closely!) and fire before being fired upon. Computer gamers are often very adept at recognizing tiny motions in an otherwise still scene, as this is the method programmers use to draw attention in game.

⁵ BBC News, “Video Games Good For You,” <http://news.bbc.co.uk/2/hi/technology/2943280.stm>, last accessed August 2007.



Figure 5. Difficult visual target differentiation can be improved through computer gaming in America's Army.

Target prioritization – Another visual skill that translates well to a military proficiency is the ability of gamers to quickly and correctly prioritize targets based on visual acquisition. This provides a significant advantage to a military member placed in harm's way during a potential conflict. Computer gamers are more adept at determining if a target is a friend or foe, and which targets are of a higher threat or value with a quick glance, as in Figure 6.



Figure 6. Computer gaming can help in assessing target threat prioritization, as in America's Army.

Ability to work within a team using practice and rehearsal – Game projects such as Full Spectrum Warrior and America's Army have attempted to capitalize on this skill that gamers develop as a by-product of attempting to maximize their competitiveness within the game. Elaborate clans, guilds, and other social networks are routinely created in order to collaboratively conquer some of the most difficult game content. These clans and guilds often have complex operating mechanisms during game play that need to be rehearsed and practiced. This skill of teamwork and team communication is something the military has shown great interest in building within its members, as it is a vital skill employed by its most effective units and soldiers.



Figure 7. A soldier confronts multiple targets in the field in America's Army.

Willingness to take aggressive action – The ability to shoot at human targets is not the only aspect of combat that video gamers have shown the ability to learn, as in Figure 7. The ability to take any manner of decisive, aggressive combat action can be taught through gaming including crossing enemy fire, moving through dangerous zones, and other aspects of combat. This was one of the goals of Full Spectrum Warrior.

Games allow increased participation – Although the best learning environments are highly interactive whether they are inside or outside of virtual environments, serious games have an additional benefit in that they allow the student to not only be an educational consumer, but also an educational producer. Students and users can modify portions of game engines or the content of the environment and world in which they are participating in. Some games with well developed tool kits and map editors have managed to live more than one life by allowing users to modify game maps, content, and game play. StarCraft, launched in 1998, is the third best selling game of all time and in Korea is played professionally in televised matches due to the ability to modify and

create new maps for game play. This further reinforces learning by effectively allowing the user / student to become the game designer / teacher.

6. Game Development

Fortunately for these disillusioned students, an emerging coalition of engineers, academics, instructors, the US military and game developers are aware of the enormous potential of learning through gaming. The simplistic mix of education software and entertainment that used to be sold as the leading form of edutainment has been replaced with stylishly developed software that seamlessly blends gaming with education, or often veils the education benefits of the game itself completely. Today many of these instructional simulations receive budgets and development resources that are many orders of magnitude larger than the earliest edutainment programs. An example of this is the America's Army project. The original America's Army game allowed civilian youth to train and experience life as an American soldier, making it both easier to recruit him and making him a more talented recruit at the same time. Over 17 million users have registered on the America's Army website, with over half that number completing "virtual basic training". Recognizing the almost perfect overlap that exists between the gamer and the military member, the US military now uses over 50 different video and computer games to reach their target population at their Department of Defense (DOD) gaming website.⁶ And the US military isn't the only organization that has realized the benefit of this largely untapped resource. Microsoft has founded the "Games-to-Teach" project. "Serious Gaming" conferences, initiatives, books, and websites have emerged.

In 2003, the global education and training market was estimated at \$2 trillion. The International Data Corporation (IDC) predicts that by 2008, 40 percent of all US companies will have serious games in their development efforts. According to Michael the Army, a leader of utilizing software to train, already spends \$7 billion annually on training in virtual environments and simulations.

⁶ Department of Defense Game Developers Community, <http://www.dodgamecommunity.com>, last accessed December 2007.

a. New Opportunities

The retail video game industry has historically been driven by technology, either the latest hardware and graphical advancements or software and the latest and greatest game engine. This has made it difficult for game developers since they've often had to discard their previous work and recreate it for the next generation of hardware, software, and customer expectations. And while there have been some stabilizing influences in the technology used by video games, the content required for games at launch has grown. It is not unusual for the team of artists, modelers, and musicians to be five or even ten times the size of the programming team. Since the content created is usually tied to the technology in a particular game, much of it is not useful for subsequent development efforts. However, serious games, simulations, and MMOGs are often designed and based on implementations that will not change with the next console release or game engine update. Graphics and model development in these games often take a back seat to accuracy of physics calculations, accuracy of stochastic outcomes, use of deterministic results, or improvements in the game engine or simulation itself.

b. Reusability

With such a thin margin of profitability for the development of one game or simulation, software developers have turned to finding more than one use for their product. Game developers often create an internal team whose sole purpose is to seek ways to re-use what was created for the original retail product. However, for the military and government, this can be impossible since they often purchase only the end product and do not own the source code used. The government is often locked into single developer contracts that often do not serve them as fully as an open source initiative will, or does not even allow them to reuse their own product in another area if a secondary use is found without adjusting the terms of the contract.

7. Bad Publicity

Despite these positive findings, research, and the cries for help of the students, most parents and educators still think of games and gaming as frivolous and a waste of

time at best, and actually detrimental to learning and harmful to the student at worst. The press aggravates this misconception with headlines about “killer games” and gaming. To counteract the prejudice against gaming, some of today’s educational games instead emphasize positive characteristics using terms such as “simulators” and “virtual environments”. Only recently has there been a change in public opinion in some segments of society about the positive aspects of gaming. An online catalog of Social Impact Games⁷ lists over 100 examples of positive gaming in fields such as military strategy, environmental science, history, etc.

C. CONCLUSION

Video games are beginning to take on new challenges in education and training with many new skills being learned just through interaction with the media of these games. These games are no longer tools to be used only for entertainment. By developing simulations and virtual environments that educate as well as entertain, civilian and military organizations can create educational tools that improve and train desired skills not taught through conventional teaching methods. These serious games, massively multiplayer games, and social networks are likely to provide key insight into development, networking, and design of future educational simulations. Adopting the positive aspects of serious games and MMOG environments is a productive way to further grow the capabilities of online large-scale virtual environments.

⁷ Social Impact Games, <http://www.socialimpactgames.com>, last accessed December 2007.

III. RELATED WORK

A. INTRODUCTION

This chapter explores serious games and analytic projects that are already in development that highlight some of the key useful aspects that games can bring to training. This includes the America's Army project for its wide appeal, game-like nature, and focus on the human individual viewpoint within the simulation. It also covers the predictive focus of the Metaverse Roadmap, the idea of linking game and simulation environments to tear down the walled gardens of social network applications and other online applications to create a truly linked Multiverse. Lastly, it examines an applied example in the form of the SAVAGE library content.

B. ARMY GAME PROJECT

The America's Army game was developed as a recruiting tool, to be played by the civilian population at large. This allows anyone to have an entertaining experience, as well as a chance to “play soldier.” By allowing civilians to play soldier, game designers have attracted more recruits to the position, as well as increased the overall proficiency of that incoming recruit. Many recruits show up to Army basic training in Fort Benning, Georgia completely familiar with the layout of the base and the training schedule they will undergo from having completed the training virtually.

Existing soldiers experience benefit from the simulation as well, by learning mission rehearsal, intelligence skills training, first aid and survival training, and other important skills for survival. To an enlisted squad leader preparing for a mission, this game may be an important part of his training as in Figure 8. Michael says in *Serious Games* that “One person’s training simulation is another person’s game.”



Figure 8. A user of America's Army is involved in urban operations.

The main goal of America's Army was verisimilitude, to be the next best thing to being a real Army soldier. And although some typical video game shortcuts had to be made to convert the real experience into a digital one, the game still has an undeniable real and gritty feel to it. Even if this realism dissuades some potential recruits, the Army still wins. Michael states that previously 13.7% of new recruits dropped out of basic training, costing the Army about \$15,000 per failed recruit. By providing a simulation of soldier life from recruit training and beyond, the simulation has recruited people more likely to succeed in the army and finish boot camp. America's Army participants, originating from over 60 countries, have participated in over 200 million hours for a total of 3.5 billion rounds of online play. The game has been downloaded over 40 million times, and spawned over 1100 fan based websites dedicated to the game.⁸ The game is available for play on Windows, Mac, and Linux and soon a coin operated arcade version

⁸ Kotaku Games, "America's Army Turns Five," http://kotaku.com/gaming/hooo_ya/americas-army-turns-five-273646.php, last accessed December 2007.

is to be launched. This arguably makes America's Army one of its most important simulations, yet is typically still thought of only as a game.

The America's Army project was built using the Unreal gaming engine, a commercial platform with good expandability. By developing plug-ins for other training avenues, the initial America's Army development can be reused again and again. An example of this expandability is the TALON robot trainer used to practice the defusing of Improvised Explosive Device (IED). Developing TALON as a plug-in to America's Army as opposed to a standalone application allowed it to be rapidly developed in just under 3 months, and at a very affordable development cost of \$60,000. As a stand alone application it would have taken many more months and tens of thousands more dollars to complete.

This concept of placing a serious simulation upon existing gaming architecture is further developed throughout this thesis. The reason America's Army serves as a case study for the creation of an X3D based virtual environment set in a mirror of the physical world is that this new MUVE can also be modularized for multiple uses. With sufficient security measures and confidential information restrictions, portions of the virtual environment can serve as training aids for existing military members. Simulation tools to explore possible outcomes of scenarios, collaborative work spaces, and other applications such as recruitment are feasible. The tool itself can place potential recruits or transferring service members side-by-side with military members they might eventually replace due to normal personnel turnover. The open-standard stability and royalty-free nature of X3D means that such web-standards based worlds can expect to enjoy great long-term persistence, growth, and interoperability.

C. METAVERSE ROADMAP

1. Background and Introduction

What happens when this generation and the upcoming generation of video games meet the newly developed “Web 2.0”? When the free simulations such as Second Life and There! are set in the backdrop of the real world through a digital model such as

Google Earth or X3D Earth? When persistent game physics are developed and enhanced to provide “good enough” approximations inside free gaming applications? The Metaverse Roadmap attempts to predict the outcomes of these inevitable transitions, as well as forecast what the coming 3D web applications and roles in our daily lives will be in the near term (approximately 10 years) and speculate even further about the Metaverse twenty years into the future. Fortunately, both of these goals are living goals adjusted by new developments and advances, and extended interaction with other online communities through blogs, wikis, podcasts, and other media. This helps to keep the Metaverse initiative current with changing hardware and software throughout the World Wide Web.

2. Bases for Speculation

The Acceleration Studies Foundation (ASF) and its partners define the Metaverse as a convergence of 1) virtually enhanced physical reality and 2) physically persistent virtual space. It is a fusion of both, while allowing users to experience it as separate entities or a conjoined reality. This is an important definition, since most of the applications analyzed in this thesis are typically only 1 of the 2 above criteria (typically the physically persistent virtual space) such as Second Life and Olive, but are not effective at providing a virtually enhanced physical reality. This failure to effectively provide a virtually enhanced physical reality is important enough, and powerful enough of a tool, to warrant the development of an application that *is* capable of meeting this Metaverse description. For example, the creation of a mirror world that contains the geometry, bathymetry, and telemetric data for a coastal city within an area of operation that is constructed and continuous updated by all sensors in the deployed space (either manned or unmanned) would meet the 2nd definition of the Metaverse and makes for a very potent military application. But the ability to overlay that data back from the virtual world to heads-up displays (HUDs) and head mounted displays in an augmented reality system for forces in the field is an ability to potent to overlook. Currently some of these applications do already mesh well with the 2D web. However, they are not sufficiently developed to gain the unique advantages of an augmented virtual reality when the tools either exist now to do so, or are available just over the horizon.

3. Enabling Trends

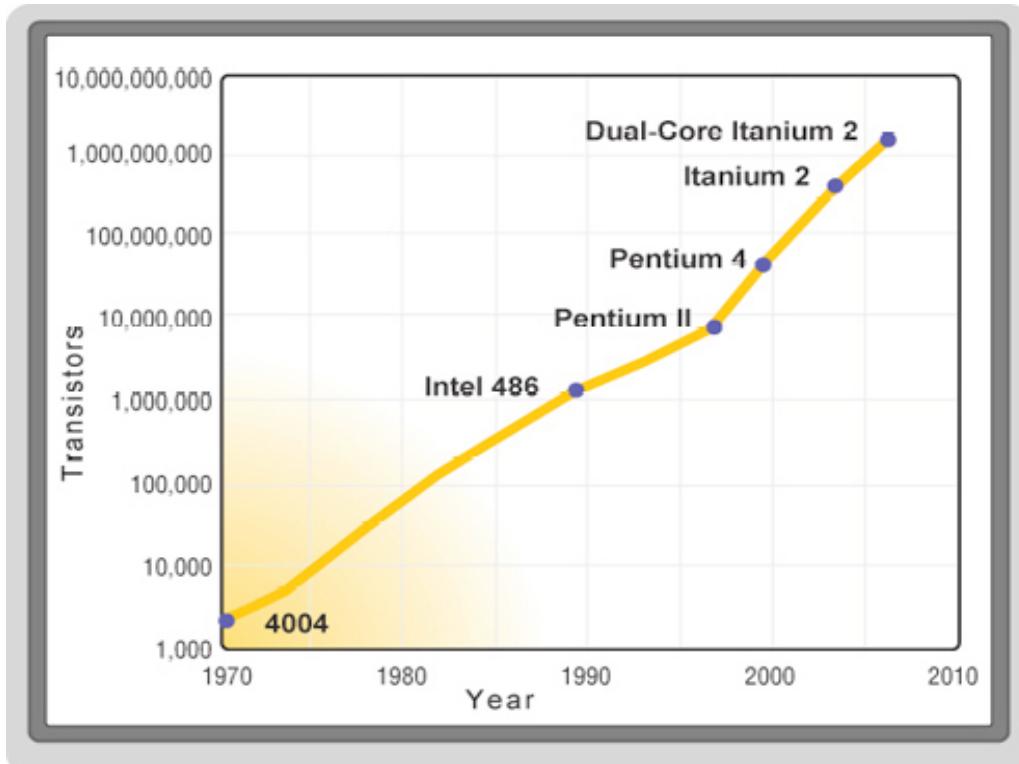


Figure 9. Moore's Law depicts an exponential increase in processing resources [From Metaverse Roadmap].

How does the Metaverse Roadmap comfortably anticipate some of these advancing changes? It can do so because the great majority of these technological trends continue to accelerate in a predictable fashion. Figure 9 is the most famous case of this accelerating trend of computing power: Moore's Law.

Moore's Law is an empirical observation that states the computational power of today's computers effectively double every twenty four months. However, Moore's Law is only one of a larger family of computer trends that are following this accelerating growth curve. Pictured in the following figures are acceleration curves showing the expansion in capacity and capability for storage, bandwidth, learning, hours logged in a simulation, and others.

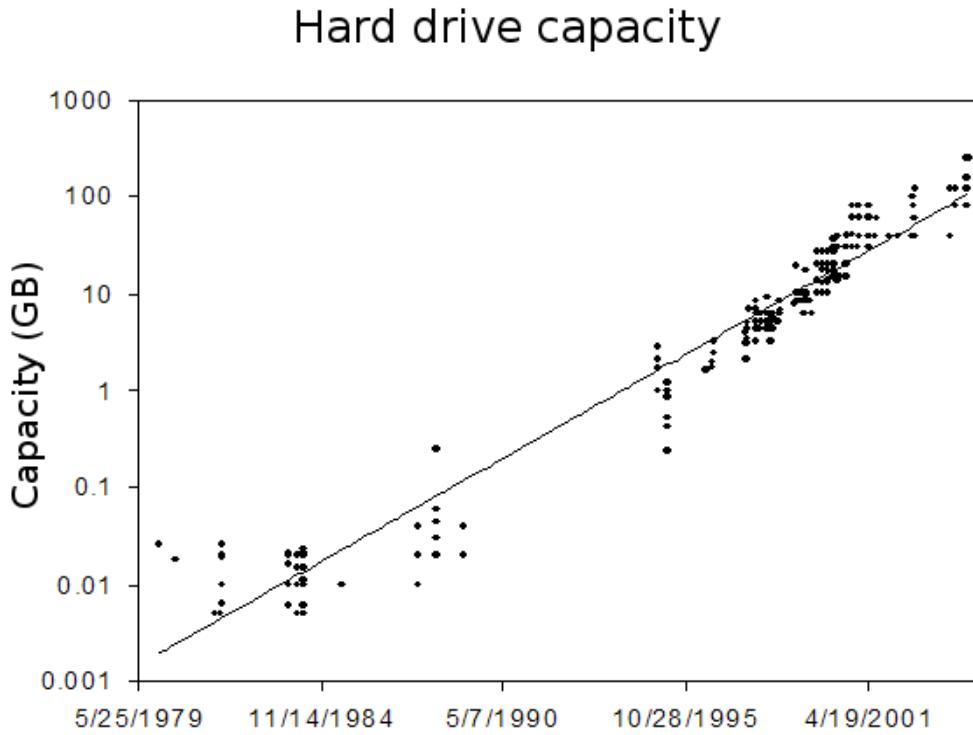


Figure 10. Kryder's Law predicts exponential growth in computer storage capacity over time [From Metaverse Roadmap].

Similarly, Kryder's Law, Figure 10, depicts the growth in computer storage capacity as a linear increase over time. Currently computer storage is not a frequent limitation of deploying most software projects. In the case of developing a persistent mirror world simulation as described by this thesis, the expected Linux cluster architecture to be used would have sufficient attached storage capable of meeting these particular criteria of use. However, forward deployed systems continue to evolve in portability, which often means ever decreasing profiles that *do* necessitate ever increasing storage capacities in smaller footprints.

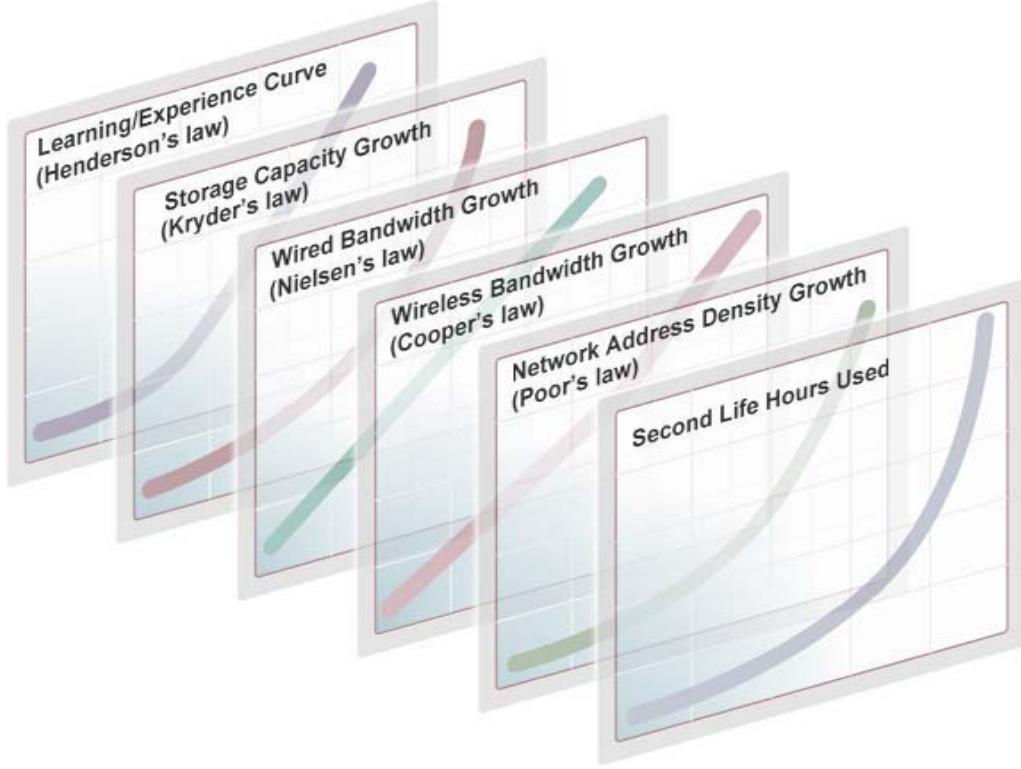


Figure 11. Exponential growth in most areas of computing resources is predicted [From Metaverse Roadmap].

Most enabling computer trends either progress linearly or faster than linearly over time. The conclusion is that the computing resources will be viable, the desire to use the 3D web for these purposes will be present, and a sufficient resource to fill the void must be developed. Note that even computer usage of simulations tools such as Second Life currently follow this exponential increase over time, as in Figure 11.

Furthermore, another advancement in the computer industry holds incredible promise for this virtual mirror world of simulation and training: unmanned vehicle data gathering. With real time input from unmanned vehicles inputting appropriately models and buildings into a virtual world, the 3D battle space can very quickly and efficiently be virtualized for simulation, training, or for information to conduct the mission. Not only are new advances being made in the automation of unmanned vehicles, but the data being gathered by those vehicle's sensors is being automated into inputs to construct visual outputs of the data. The SAVAGE library provides an excellent example of using

models in isolated stand alone contexts up to complex scenarios such as the AUV Workbench project (<http://savage.nps.edu/AuvWorkbench>).

4. Lifelogging.

Another key aspect analyzed by the Metaverse Roadmap that the military has shown desire to develop is the “lifelogging” concept. This is the concept of tracking real world data within a virtual environment or data base that opens up vast new dimensions of usability for Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR). By allowing key signals to be tracked in the real world, such as pulse, other vital signs, position, viewpoint, and other key data, the virtual environment again becomes a two-way functional tool of communication. Imagine an operation where an Unmanned Aerial Vehicle (UAV) is sending real time data to a simulation that is based upon the real world geometry and terrain of the operating environment. A unit conducting operations is in the area receiving data and recommendations from the virtual environment as well as passing back their position and current status. Lifelogging will allow enhanced capability within the environment, as well as a means of playback to analyze what occurred to improve future actions. This can be useful to help determine if improper firings occurred, why an ambush was not detected, or how best to proceed to win an imminent conflict. How long before lifelogging devices are implemented on soldier’s weapons?

D. MULTIVERSE AND SOCIAL NETWORKING

Similar to the anticipation of the Metaverse Roadmap project is the Multiverse. The Multiverse is a system of linked massively multiplayer games that allow a user to bring their avatar identity through any system of interrelated MMOG. This ability to interlink game environments is a good example of what the Metaverse project anticipates the future of gaming to be, as well as is a representation of the type of project creation this thesis argues for.

The Multiverse project is similar to a Software Developer’s Kit (SDK) to build the back-end on the game architecture, as is used at the other MMOGs. However, the

engine is open source to allow programmers to create their own content and environment as they see fit. This Multiverse concept cuts down on development cost, which in architecture such as a MMOG, can sometimes skyrocket to \$20 million or more.⁹ Multiverse claims they can get small businesses going in the MMOG arena for as little as \$10,000. The software used to develop a world within their multiverse is free, with Multiverse charging a commission on subscribers that eventually join the developed world.

In order to prove the Multiverse project can work, the developers created their own demo world using their software. The project “Kothuria” is the result of this development. The graphics are not comparable to the artist rendered creations of the large company’s such as Sony’s Everquest II, but by allowing the physics and artificial intelligence to be layered Kothuria can keep current with the larger projects as other open source developers make improvements to these modules.

By lowering the initial cost of investment and charging fees based only on subscription, the Multiverse, like Second Life, holds great potential for the academic development of a virtual world for experimental purposes without a huge dollar investment. It is not as full of a solution to these questions posed by this thesis as other solutions are, as will be shown.

E. SAVAGE X3D WORLD MODELS

HTML (and its derivatives) is the language of the World Wide Web. By itself, HTML can depict a webpage whether online or offline, but when those pages become hyperlinked the whole experience of the web is utilized. Similarly, the Metaverse will need an open source, extensible, 3D language that is capable of depicting a single model, or is capable of being used to flesh out entire worlds, whether online or offline. X3D is well positioned to be that open source standard, and the SAVAGE library¹⁰ is an example

⁹ Cnet News, “New Gateway to Multiple Virtual Worlds,” http://news.com.com/New+gateway+to+multiple+virtual+worlds/2100-1043_3-5983264.html, last accessed December 2007.

¹⁰ Scenario Authoring and Visualization for Advanced Graphical Environments, <https://savage.nps.edu/Savage>, last accessed December 2007.

of how that language can be used. A fully persistent 3D mirror world, as proposed in this thesis, is just one instantiation of use for the language of Web 2.0, as a traditional website is a discreet unit of the existing web. As two websites will have differing designs, bases, and dimensions, so two will any 3D applications differ. The important element is that the language be scalable and interlinked to accommodate a simple, stand-alone model isolated to an individual computer up to being the format for linked game universes to form a Metaverse.

The SAVAGE library uses the X3D standard to create models with primarily military-based applications. It is remarkable in that complex and detailed models can be created by professional modelers as well as students and amateur enthusiasts, or simple models that “just get the job done” can be created easily as well. Figure 12 is a model developed by Major Charles Lakey (a student modeler) using X3D-Edit¹¹, is freely available in the SAVAGE library, and is an excellent example of the capability of the standard. These models are extensible, open source, and as X3D is based on XML, are easily parsed to and from scripts and engines. Furthermore, being XML based, it can easily be imported and exported to most existing modeling programs. This makes it ideal to convert these existing models into other formats for use in proprietary programs, such as converting an X3D model into OLIVE format. It is also ideal for taking existing models in popular formats such as .3DS (3D Studio Max) or Maya and converting them into X3D using a variety of software such as Flux Studio or the RawKee plug-in¹².

¹¹ Web 3D, <http://www.web3d.org>, last accessed December 2007.

¹² Sourceforge, search “Rawkee,” <http://rawkee.sourceforge.net/>, last accessed December 2007.

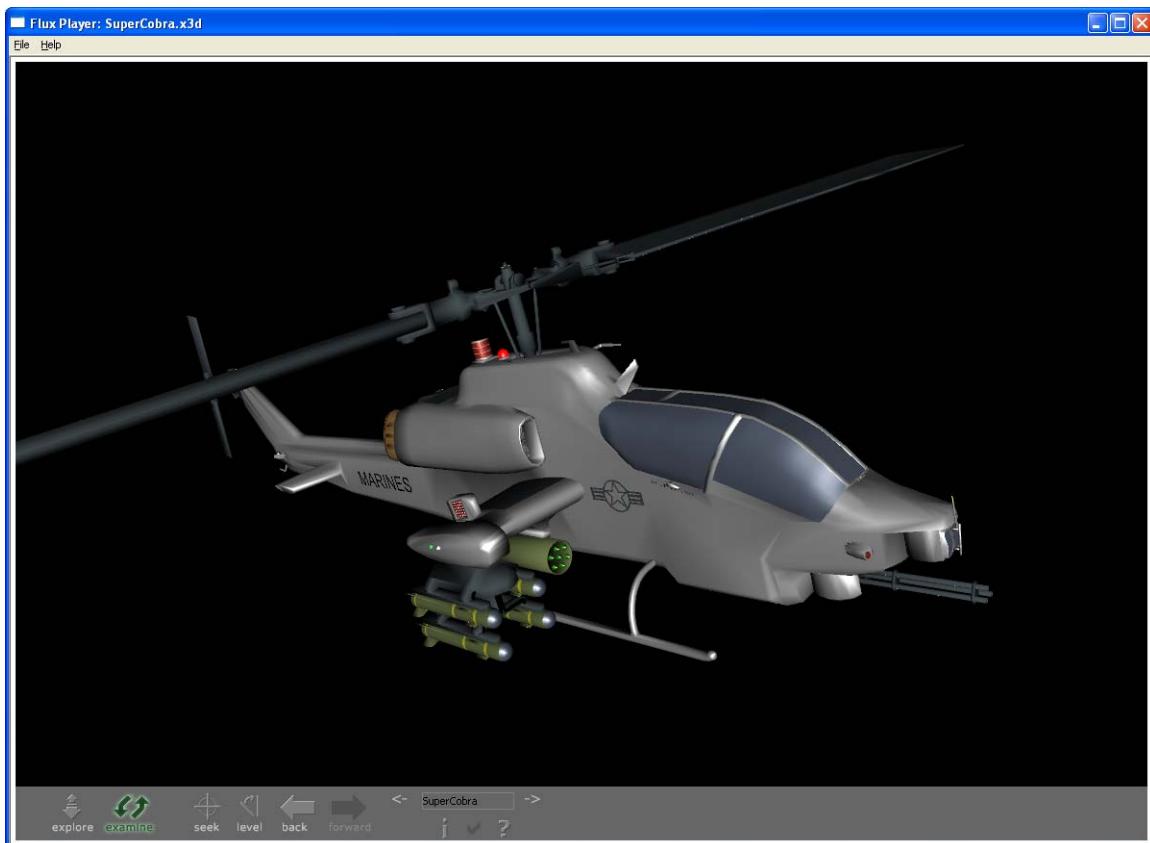


Figure 12. Amateur .X3D model is displayed using Media Machines Flux Player.

Lastly, its use of metadata provides a “self-awareness” that would be ideal for military use. Keeping hundreds of thousands of users with very differing levels of network connectivity and desired uses for the language current is not a simple task. Rauch demonstrates in his thesis *Savage Modeling and Analysis Language (SMAL): Metadata for Tactical Simulations and X3D Visualizations* that these virtual environments can be automatically generated and kept current and synchronized. Nicklaus demonstrates this concept of automatically generating a 3D world for visualization of data transferred in an XML format in a military context in his thesis *Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE)*.

F. SECOND LIFE FOR RESEARCH

Commercial endeavors are not the only interested party in these newly developing, freely available virtual environments. For research purposes, these environments can provide a unique atmosphere to learn how these environments are built and managed by system administrators, as well as how they are used its player base. These virtual environments provide a free glimpse into a microcosm of society that might not exist in any other fashion, and researchers in law, economics, and social studies have taken notice.

Specifically, the Research Roadmap provides a report based on findings of a study Group at the Columbia Institute for Tele-Information and The Conference Board. According to Mennecke, the purpose of the study group was to examine research within virtual worlds along a wide-ranging set of issues including: 1) technology challenges 2) business strategy challenges 3) political and governance issues 4) consumer acceptance issues 5) cyber-crime and security issues 6) legal issues and 7) national security issues. Several of these research areas are by products of the virtual environment itself, and provide a means of increasing returns.

G. CONCLUSION

The Metaverse Roadmap attempts to provide a predictive look at the future of Web 2.0 to better allow companies, educators, developers, and users prepare for the future. Other applications are analyzed, such as the America's Army game project, social networking applications, the Multiverse, the Research Roadmap, and the SAVAGE archive, that contain key features and elements related to questions asked by this thesis. These applications provide insight into desired features, usability and extensibility, a roadmap to development and research, and other lessons learned from developing software of this type that apply directly to the creation of a fully persistent 3D simulation that uses a mirror world environment.

IV. 3D GRAPHICS INTEROPERABILITY

A. INTRODUCTION

This chapter analyzes the approach some of the industry leaders such as Makena Technology's There!, Forterra's OLIVE, and Linden Lab's Second Life are taking at 3D model formatting. Advantages and disadvantages of their model formats are analyzed, and what the results of their model choices mean with regards to a fully persistent mirror world simulation as described by this thesis. In the cases where propriety model formats are used, the possibilities of model conversion into other formats are analyzed.

B. MOTIVATION

One of the key factors affecting game development of any virtual environment is the choice of 3D model representation. Most programs develop proprietary means of model construction inside the game engine, which is advantageous to the game developer to tailor the model to interact fully with the game engine. This is not a significant issue if the game does not need to interact with any other virtual environments. However, this proprietary architecture approach will not work in a Metaverse model of strongly linked virtual environments, which is where the true power and effectiveness of these simulations reside. Further complicating the proprietary architecture issue is the relatively new development of in-world modeling tools for the instant creation of persistent real-world geometry and terrain. The ideal solution is to develop models using a format that is capable of being rendered quickly by a game engine; is royalty free and open-standards based to interact with any other virtual environment; is sufficiently simple to allow amateur user creation of models but also provide sufficient level of detail for professional model design; and lastly the model itself should be stored in a file format that allows for in-world manipulation, more robust external model creation, as well as allow importing and exporting between the most popular model creation software

programs. Because X3D is an XML file and is an open standard, with an established and growing community behind it, it meets all of the above requirements and stands out as the ideal choice for model use.

C. 3D MODELS

1. OLIVE

Of the simulations studied, OLIVE provides the easiest means of developing content for use inside their virtual world. The software package that ships with the baseline OLIVE platform include importers that work with Autodesk 3DS Max to import all forms of geometry, as well as some animations and terrain files. It also comes with an offline Scene Editor for developing a unique environment and a standalone Model Viewer for artist's development of models, but at the time of this thesis does not allow for in-world model creation. It has an Emotion tool to allow blending several of the base emotions into longer more complex composite expressions. Further customization and features will be achievable with the release of their SDK.

The actual model format used within the game engine is itself proprietary and was not available for stand-alone analysis. Translating existing X3D models into OLIVE is a 2 step process of first loading the model inside 3D Studio Max, converting the file to .3DS format, then importing that .3DS file into OLIVE's environment. The net result is that getting existing models into OLIVE is very convenient and easily accomplished, but removing content from OLIVE or networking the OLIVE simulation to another simulation can not be accomplished without modifications to the simulation and Forterra's consent.

2. There!

The model development process is unique for Makena Technology's There! compared to the other simulations and games studied in this thesis. Firstly, models are not created in-world, but are imported after being created externally, and then the content must be approved by Makena developers. This enables the world managers to check

consistency to better create uniform worlds, not only in appearance but in the geometry and design of the models. Models can be created using authoring tools such as Autodesk’s Gmax or 3DS Max. They must have roughly the same level of details as the provided sample models, with all textures void of any copyrighted material. The model is then submitted through a paid process before a game developer approves the model for in-world use. Once in-world, the items can then be auctioned off to other There! participants. As with OLIVE, this means X3D models must first be converted to .3DS files, with appropriate geometry and model constraints, before being imported into the There! environment.

3. Second Life

The Second Life model format is not only proprietary, but is also truly unique. This unique model format was used to allow for smaller network transmissions, but result in a model format that does not work or cooperate well with other environments and programs.

Models created in Second Life are built from 3D geometrical shapes called “prims” (short for primitives). Each region can support 15,000 prims (with an additional 10% resource available to allow for trading / creation). Each region is 65, 536 simulated square meters. Initially one server held one entire region, prompting residents to refer to regions as sims – short for simulators. Now every server holds two regions, but the name has stuck.

Prims aren’t truly primitives, but are instead parametric models (3D objects with a path and a profile). The profile is swept around the path through space to create a solid. For example, a sphere is a semicircular profile with a 0 radius path. The path can be a 1D, 2D, or 3D curve in space, while the profile is usually 2D (but in some cases can be 1D, 2D, or even 3D in certain rare geometries). This is done to minimize the amount of data that needs to be shared between the client and server in order to allow Second Life to operate on older machines with slower internet connections as well as broadband-connected machines with high-end graphics cards. In effect, Second Life is passing the information to describe *how* to draw the geometry as opposed to the geometry itself.

However, this style of model creation means that some useful and routine object manipulations, such as vertex manipulations, can't be performed on Second Life prims. This also currently prohibits manipulations such as pixel or vertex shading, which results in the cell shading look prominent with Second Life that is not seen in other simulations. This proprietary method severely reduces extensibility and interoperability of Second Life.

Importing existing 3D models into Second Life can be difficult, and is possible only through a several-step procedure, which increases the chance of data corruption and loss within the model itself. Although X3D supports both polygonal geometry and parametric geometry, there still exists no way to directly create a parametrically defined X3D model in Second Life using third party modeling software, although this functionality could be developed.

a. Prim. Blender

The first option to incorporate an X3D model into Second Life is the Prim.Blender project (spoken “prim-dot-blender”). The Prim.Blender project is an open source tool for the Blender Application Programming Interface (API), written in the Python scripting language and freely available from Sourceforge.¹³ The python script allows authors to develop the primitive geometry inside the Blender tool and then save a project as a .prim file. This approach does not allow authors to use the full power of Blender to import existing 3D models, then convert them to primitive files, but does allow them to create models in a much more user-oriented design environment, as well as creating objects while offline from Second Life. The display shows the following primitives: Prism, Sphere, Torus, Tube, and Ring. Not shown are cubes and spheres, which Prim.Blender is fully capable of creating. In addition to all the routine translations, rotations, scales, and 3D object manipulation capable with Blender, the Prim.Blender project also allows an author to natively assign material values to created geometry. All the material properties available in Second Life (Stone, Metal, Glass,

¹³ Sourceforge, search “Prim.Blender,” <http://sourceforge.net/projects/primdotblender>, last accessed August 2007.

Wood, Flesh, Plastic, and Rubber) are available in the pull down menu. The figure below shows Prim.Blender version 0.3.0 (version 0.4.0 is the latest version) running in Linux.

Rezzing the constructed model inside Second Life is accomplished through the clever use of a Second Life note card import script developed by Jeffrey Gomez. By wrapping the created .prim file in the block script, then calling the rez script (equivalent to a “cut” and “paste” operation) from inside Second Life, modelers can import geometry into the Second Life grid.¹⁴

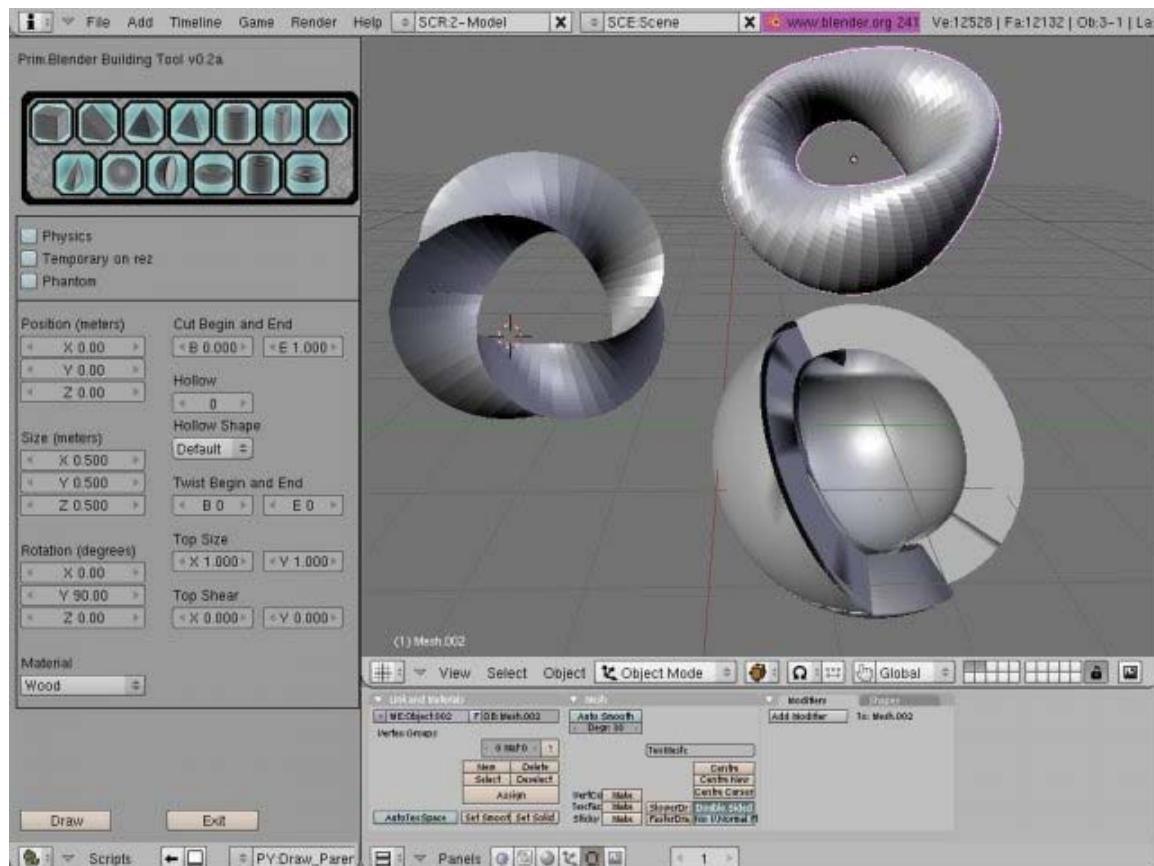


Figure 13. The Prim.Blender second build in Mandriva Linux. 2.16 kernel is used to create basic geometry.

¹⁴ 3pointD, “Sketchup > Blender > Second Life,” <http://www.3pointd.com/20060626/sketchup-blender-second-life/>, last accessed July 2007.

Figure 14 shows the latest version of Prim.Blender running on Windows XP. Version 2.42a of Blender will compile and run the script without editing, but the latest version of Blender (version 2.43) requires editing of the python script prior to compiling and executing (changes necessary are annotated in the Sourceforge Prim.Blender forum under “Help”).

Although this tool cannot import existing 3D models into Blender, and then convert them into .prim files to be rezzed within Second Life, it does allow for much quicker creation and editing of primitives compared to the in game model creation software. At the time of this writing, Linden Labs has announced that an export tool for Maya is soon to be released; with follow up exporters for 3Ds Max and Blender.

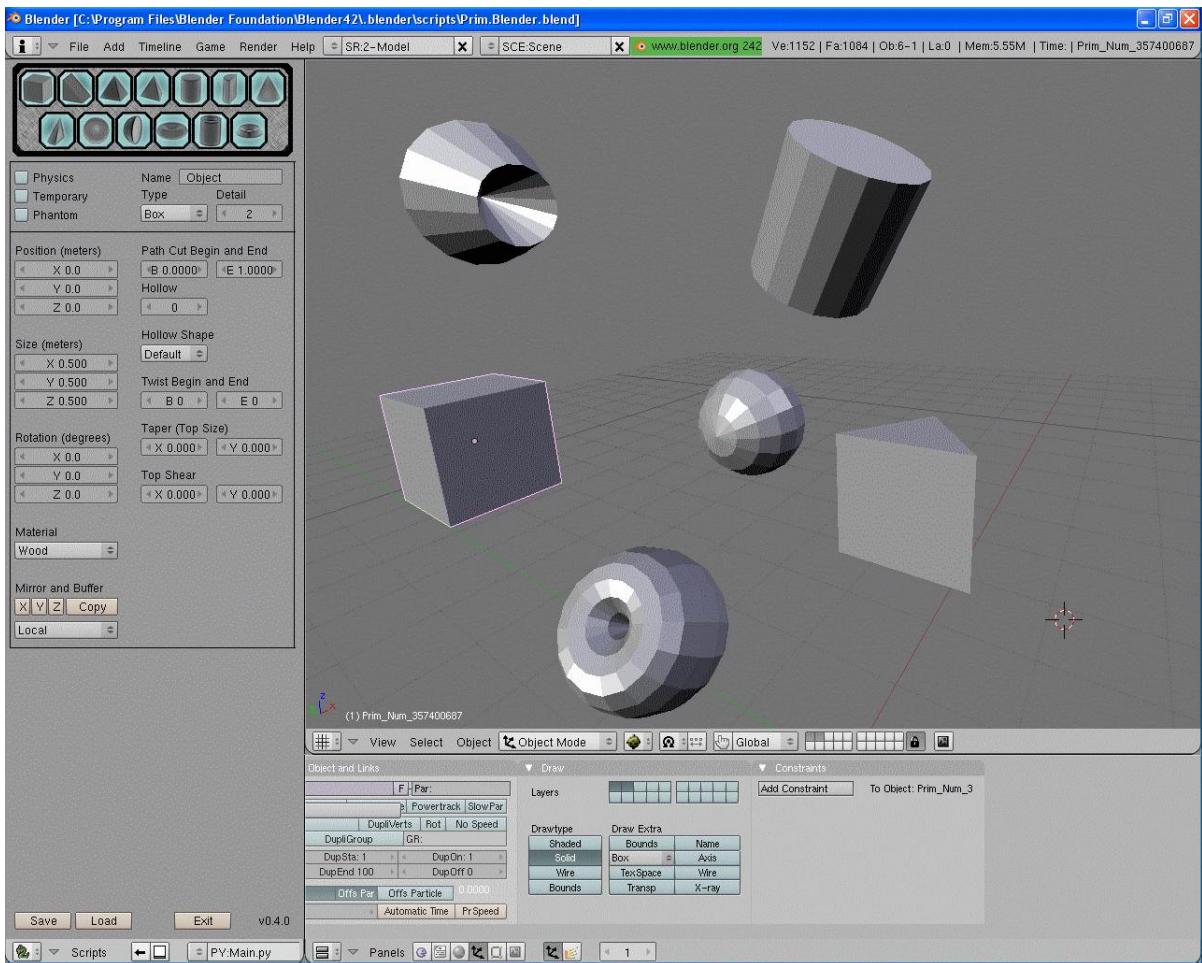


Figure 14. The Prim.Blender version 0.4.0 on Blender 2.42a (running on Windows XP OS) shows the basic primitive object types.

*b. **OBJ Importing Script***

Second Life resident Jeffrey Gomez has developed a freely available script for the use of model importing into Second Life from the .OBJ file format. The .OBJ format is used since it is a popular format capable of importing / exporting in Maya, 3DS Max, Blender, Milkshape, AC3D, and many others. It is also used since the .OBJ file stores the model data in readily parsed format. Each vertex, normal, and texture coordinate is stored as a list of points in 3 dimensional space (e.g. v 0.123 0.234 0.345). Each face is then given by a set of indices with the vertex / texture / normal coordinate array.

The LSL Import tool macros this list of vertices into a triangle primitive inside Second Life. This means it creates at least one triangle primitive for every face on the imported model. This quickly creates a primitive model that is prohibitive in size, and does not work for all but the simplest of geometric shapes (geometry on the order of 100 vertices or so). However, no other means of model importing exists at this time.

*c. **Maya / 3DS Max importing.***

As mentioned previously, Linden Labs has announced that they will incorporate an exporter for Maya, followed by additional export tools for Blender and 3DS Max. However, due to the parametric design of Second Life, Linden Labs is unable to directly incorporate model transfer via vertex to vertex. Importing, and new build tools, will come in the form of a new “sculpted prim”, that allows for creating a more complex and natural shape than is currently possible.¹⁵

4. Project Darkstar

Project Darkstar, and Sun’s demonstration version of the software, Project Wonderland, allow for native importing of the X3D format. This is the only simulation analyzed in this thesis with this ability. Since X3D is an open standard, there exists a

¹⁵ 3pointD, “Sketchup > Blender > Second Life,” <http://www.3pointd.com/20060626/sketchup-blender-second-life/>, last accessed July 2007.

variety of solutions to map models in any of the studio modeling programs such as Maya, 3DS, or even Google Sketchup to the X3D format. This means nearly any model can either be natively imported, or easily converted to the X3D format for import. Project Darkstar does not strictly define the modeling format itself, so other formats are also freely usable. This flexibility and versatility make it useful when considering networking virtual environments and social networks.

D. OPERATING REQUIREMENTS

1. OLIVE

OLIVE is designed to run on commodity Windows platforms and Linux servers. However, the program itself is well scaled to fit a variety of machine specifications so hardware requirements can be deceptive. OLIVE can be used to create global sized worlds capable of sustaining massively multiplayer simulations, or can be used by a single user in one locale.

As a minimum, a PC client should have the following requirements:

- a. Operating System: Windows 98 or better
- b. 256mb RAM (512mb recommended)
- c. Pentium III 800mHz or better
- d. accelerated graphics card with 32mb of VRAM

2. There!

Since There! is designed to be a more pleasing graphical environment than its competition Second Life, it has slightly higher graphics requirements but no support for Linux or Macintosh as of this writing. The full Windows requirements are:

- Internet Connection: Yes (broadband recommended)
- Operating System: Windows 98, ME, 2000, XP
- Processor speed: 800 MHz Pentium III or better
- Memory: 256 meg memory or better

- Graphics: NVIDIA GeForce or better, ATI Radeon 7200 or better
- Storage: 500 mb disk space

3. Second Life

Although Linden Labs has placed the client code for Second Life under an open-source license, they have announced a similar intention but not yet released the server code. This means that in order to use Second Life, in any facility, users must have an internet connection to connect to a Linden Lab's server. Second Life is also currently only ported to the Windows operating environment (Windows 2000 or higher); however, Linden Labs does provide an alpha version of the client for Linux¹⁶. Open source ports of the client to Linux have also begun with the release of the client source code. The full system requirements are as follows:

a. Windows

- Internet Connection: Yes (broadband recommended)
- Operating System: Windows 2000 or higher (Linux in alpha)
- Processor speed: 800 MHz Pentium III or better
- Memory: 256 meg memory or better
- Graphics: NVIDIA GeForce 2 or better, ATI Radeon 8500 or better
- Other: OpenGL support

b. Macintosh

- Internet Connection: Yes (broadband recommended)
- Operating System: Mac OS X 10.3.9 or better
- Processor speed: 1 GHz G4 or better
- Memory: 512 meg memory or better
- Graphics: NVIDIA GeForce 2 or better, ATI Radeon 8500 or better
- Other: OpenGL support

c. Linux

- Internet Connection: Yes (broadband recommended)

¹⁶ SecondLife Community, "Linux Client Alpha," <http://secondlife.com/community/linux-alpha.php>, last accessed August 2007.

- Operating System: Any reasonably modern 32 bit Linux kernel. No 64 bit mode available.
- Processor speed: 800 MHz Pentium III or Athlon or better
- Memory: 256 meg memory or better
- Graphics: NVIDIA GeForce 2 or better, ATI Radeon 8500 or better
- Other: OpenGL support

4. Project Darkstar

Project Darkstar is a new development and as yet does not have a release capable of having system requirements to run against. The platform works well on Linux, Windows, or Mac OS X and is quite stable despite being an early release. As the software moves towards a releasable product, system requirements will become easier to specify. For now, any machine capable of running the java virtual machine with even a low-end 3d graphics card can attempt to install the client and run the application. Most machines experimented on in this study were sufficiently capable to act as both the server and client using a local-host configuration for testing.

E. CONCLUSION

This chapter analyzes the model format choices of the leading simulations, such as There! and Second Life, as well as others studied in this thesis. It analyzes the advantages and disadvantages of the model format used by each of these simulations, and how that format choice affects the use of that particular simulation for the development of a fully persistent mirror world application as described in this thesis. In cases where a proprietary format is chosen, methods to import and export X3D to and from that proprietary architecture are analyzed.

V. NETWORKING INTEROPERABILITY

A. INTRODUCTION

This chapter takes a macroscopic view of networking for several of the simulations analyzed in this thesis, so an overall understanding of the difficulty of the networking situation can be understood. Performing a detailed, microscopic view of each of the simulations is beyond the scope of this thesis.

B. MOTIVATION

As seen in Figure 15, there are six core components of video game technology: the 3D Engine, the Graphical User Interface (GUI), the Artificial Intelligence (AI), Persistence, Networking, and Physical Models. The questions raised by this thesis analyzed simulators that provide the majority of these core components already in place for the simulation developer, with some simulations providing the source code to modify the components and some not. Developing the 3D engine, object persistence, AI, and the GUI are all vital components, but will either be performed by a development team or will use the simulation’s components as provided, and are beyond the scope of change of end-users. Furthermore, changes made to these components will arguably not significantly affect, or even be apparent to that end-user.

However, the remaining components, that of physical models and networking, affect the user in different but vital ways. Physical models will comprise the view into the virtual world for the user, and as such were analyzed in relative detail in the preceding chapter. Networking will next be examined, not because users will be able to make changes to the networking architecture, but rather any changes made to that architecture will affect every user of the world. The changes could be detrimental enough to prevent connectivity at all in certain situations, which needs to be adequately handled by the simulation: the ability to still be a useful tool while not connected to the internet, or networked with other simulations.

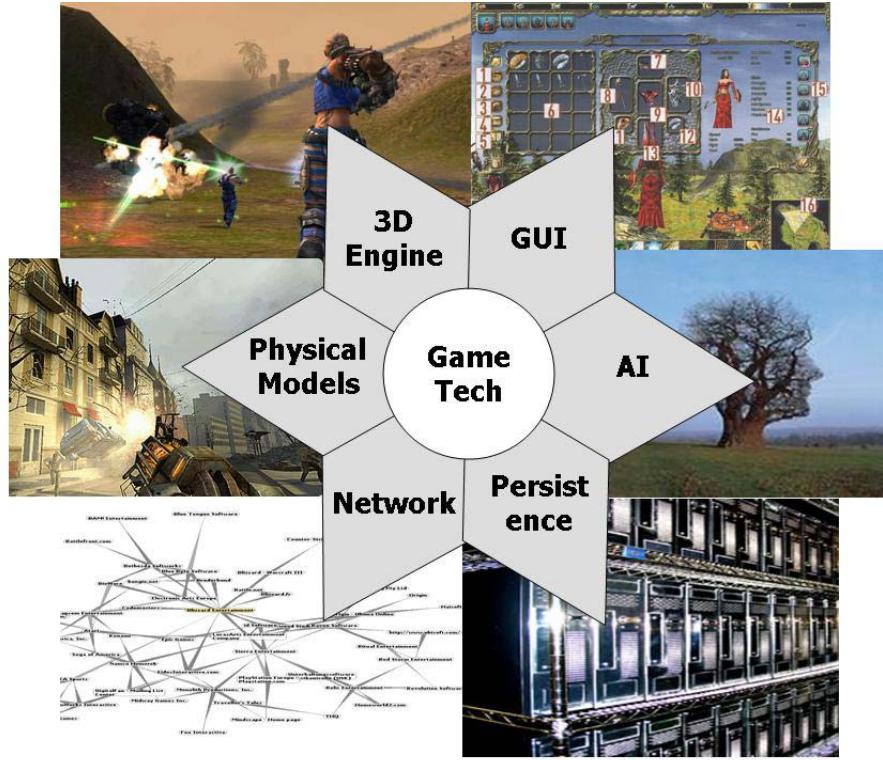


Figure 15. There are six key technologies that drive computer games [From *Serious Games: Games That Educate, Train, and Inform*].

C. OLIVE

One of the key drawbacks to OLIVE is that it requires broadband connectivity, with a minimum of 128kbps throughput. This will not be an available resource to the majority of forward deployed units and users. Furthermore, satellite broadband services are not supported due to their high latency, which is another drawback for forward deployed personnel. However, security firewalls equipped with Network Address Translation (NAT) or Internet Connection Sharing (ICS) do still usually work with OLIVE, which help to provide greater networking flexibility and connectivity.

Specifically, OLIVE connects on TCP connections 2311 to 2330 for both inbound and outbound traffic. OLIVE also utilizes ports 2311 to 2330 UDP for inbound and outbound communications. Another large advantage to the OLIVE simulation is that it is

the only simulation analyzed in this thesis that is natively capable of interacting with DIS/HLA architecture, without subsequent developer modification. As seen in Figure 16, OLIVE has attempted to place itself inline in the natural progression of simulation software to become a versatile team trainer for a variety of solutions, not just military use. Its reliance on broadband connectivity might lessen its wide scale deployment.

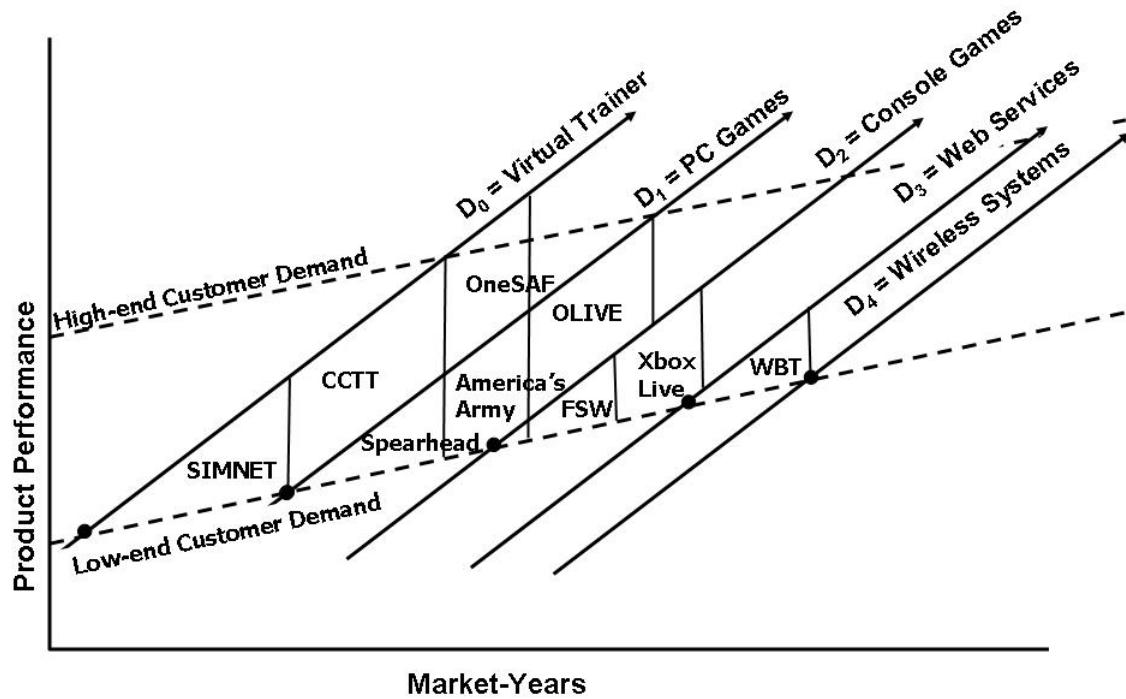


Figure 16. Simulations have progressed from professional virtual trainers to ubiquitous web applications for all users [From Metaverse Roadmap].

D. THERE!

There! is capable of running on 56k dial-up connectivity, which is a distinct advantage over OLIVE, but runs more smoothly on broadband connectivity. Almost all port traffic is passed via TCP connections for web browser making it capable to run There! behind most firewalls without security administrator reconfiguration, which is not

true of OLIVE or Second Life. Specifically, outbound TCP ports 2311, 2313, 2315, 80 (HTTP), and 443 (SSL) must be open. Inbound and outbound UDP ports 2311, 2313, and 2315 must also be unblocked and open.

There! uses compression algorithms to minimize network traffic, as expected, but these algorithms are not quite as capable as Second Life. However, There! does appear to run smoother than Second life to the user as a simple result of passing more network traffic and updating more information within the simulation per packet. Of the simulations analyzed in this thesis, There! provides the greatest network connectivity without any modification to the software.

E. SECOND LIFE

By scaling and adding servers, users continue to add to the available “land mass” of Second Life. Linden Labs owns and maintains the servers, but has announced their intentions to make the server code available. This is significant to this thesis to provide a sense of scope of the issue of developing a simulation using a mirror world background. This is an issue that truly does not scale well, and would require a somewhat different approach than typical MMOG architecture and Second Life “server farm” architecture. This trade off of terrain vs. network performance is a significant issue for an ambitious mirror world project, but is beyond the scope of this thesis.

Specifically, Second Life connects to ports 443 TCP/IP, 12035 - 12036 UDP, 12043 TCP/IP, and 13000 to 13050 UDP.¹⁷ This means the majority of firewalls default to block Second Life and must have the appropriate ports opened up to ensure connectivity, which would adversely affect connectivity from the majority of military and educational institutions. As mentioned earlier, of the simulations analyzed, Second Life makes the most of any network traffic it generates to keep the simulation updated.

¹⁷ SecondLife Community, “Network Requirements for Second Life,” <http://secondlife.com/knowledgebase/article.php?id=029>, last accessed August 2007.

F. PROJECT DARKSTAR

The network capabilities and requirements for Project Darkstar continue to shift and evolve as Project Darkstar continues development. Sun developers are purposely not delving too deeply into the networking architecture to allow game developers to more fully modify their own existing implementations through the SDK. This allows the game developers maximum flexibility. This theme persists throughout all of Sun's development decisions, making Project Darkstar the most flexible and developable simulation analyzed by this thesis. By providing the framework for the most crucial aspects of a MMOG, but not locking game developers in to just one solution, Project Darkstar has created a "format" for a MMOG that allows game developers to use existing game models, networking architecture, artificial intelligence, etc., or to move away from the default and install and configure their own components for use. In general, the system is designed for Linux cluster architecture which makes it a good choice for the development of a persistent mirror world simulation using MMOG architecture as described by this thesis. The network transmission protocols, written in Java, are fully customizable to allow nearly any format desired by game developers to be parsed by the server. This makes Project Darkstar the only fully configurable simulation, of the projects studied in this thesis, in terms of network connectivity.

G. CONCLUSION

By analyzing the network connectivity and networking issues within each of the major simulations, an overall understanding of the difficulty of the networking situation can be achieved. Networking issues are a crucial factor when discussing interlinking virtual environments, as in the Metaverse concept. Networking issues will also be a key component of designing or implementing a persistent mirror world simulation as described and defined by this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. INFORMAL COMPARISON OF VIRTUAL ENVIRONMENT FEATURES

A. INTRODUCTION

This thesis examines many social networking and simulation programs. As is often the case with software systems, no one program can contain every desired feature necessary to achieve multiple results. This is also the case with a project as large as developing a fully persistent 3D simulation with a mirror world environment. Thus the key advantages and disadvantages of each of the major simulation environments are analyzed, and recommended military and educational uses are provided for each tool.

B. OLIVE

1. Advantages

a. Visual Appeal

Although a strong visual appeal is neither a prerequisite nor a guarantee of success in a virtual world, it can help facilitate verisimilitude and a user's sense of immersion inside the virtual world. OLIVE is the most visually appealing of the virtual simulations studied in this thesis, as can be seen in Figures 17 and 18.



Figure 17. The creation of an OLIVE soldier avatar is achieved through sliders.

b. Military Collaboration

OLIVE also appears to be the only simulation with direct input from designers having extensive military experience. It is also currently being used as a platform for military intelligence collaboration, allowing Army intelligence analysts a vehicle to test evolving theories of current and emerging threats through role playing and interaction to determine the validity of a wide range of intelligence scenarios.¹⁸

¹⁸ Forterra Systems, Inc., <http://www.forterrainc.com/defense.php>, last accessed December 2007.



Figure 18. The OLIVE avatar mans a fully configurable checkpoint.

c. Voice Communication

As one would expect from a training program that sells itself as a team trainer, voice communications within OLIVE are well achieved. Directional audio, different communications channels, and clear concise voice communications make OLIVE the best choice among all of the simulations studied in this thesis for a simulation that requires native voice capability.

d. Commercial Development

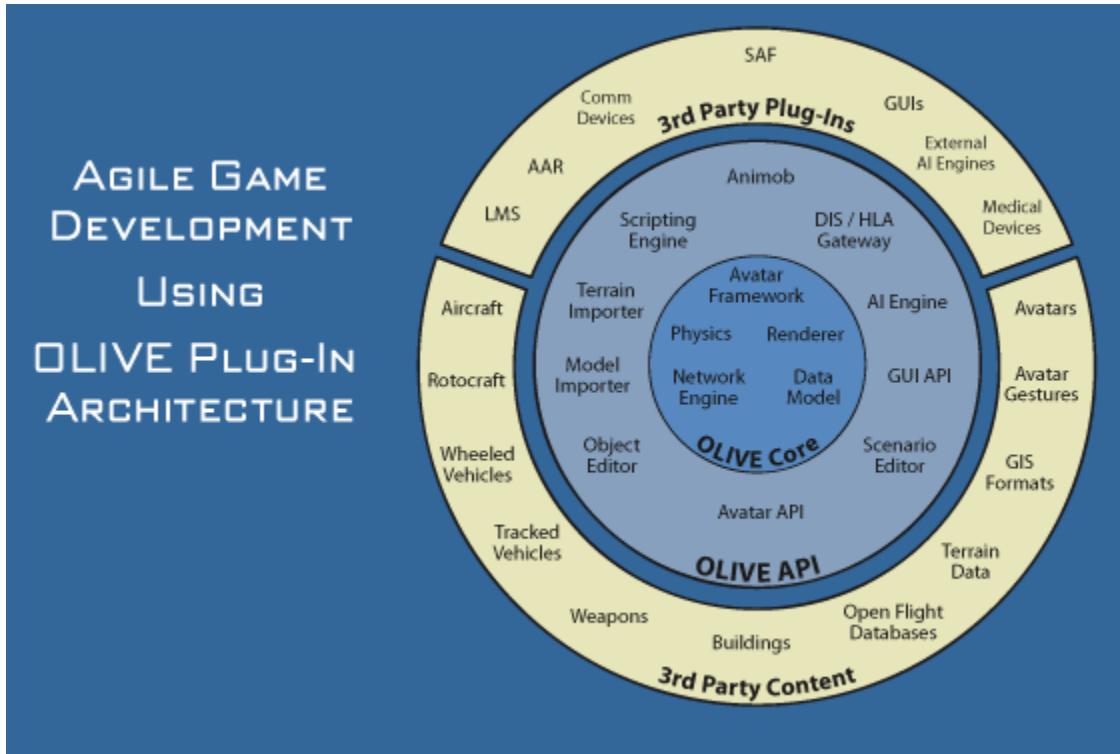


Figure 19. This figure depicts OLIVE's agile game development concept [From Forterra].

For the desired end state of this thesis: that is a fully persistent, 3 dimensional, virtual world with sufficient depth and detail to be used as a tactical and team trainer, OLIVE's commercial development has a distinct advantage. The software is well thought out and well designed, and comes with a knowledgeable support team to work with developers to achieve desired simulator goals. OLIVE also provides an SDK to help achieve individualized solutions. Changing the look, feel, and operability of OLIVE is then a process of creating sufficient plug-ins while still using the OLIVE game core and API, depicted in Figure 19, which is also a top level goal of a simulation that answers the requirements of this thesis. Nevertheless, the overall architecture remains proprietary and closed.

e. DIS / HLA Gateway Connections

OLIVE supports an optional DIS / HLA gateway to facilitate integration of its software with existing simulation networks. OLIVE has the ability to provide proxy objects for the remote simulations within OLIVE itself, which is a feature unique among the other simulations studied. OLIVE also further extends the gateway by filtering the amount of data exposed and do coordinate translation for the legacy system since OLIVE can simulate entities anywhere in the world while most legacy systems cannot.



Figure 20. An OLIVE soldier pats down a citizen for hidden weapons at a manned checkpoint.

f. Vehicles

OLIVE does provide for rudimentary vehicle use (captured in Figure 21), which is essential for any simulation destined for military use. Social network applications are unlikely to develop this functionality, and Second Life and There! use vehicles as a form of entertainment, so utility is not a prime consideration in the deployment of these vehicles.



Figure 21. One of the strengths of OLIVE is its use of vehicles.

2. Disadvantages

a. Commercial Development

OLIVE is a commercial product developed by Forterra technologies and must be purchased and maintained through them. This is both an advantage and

disadvantage, as it is with all purchased software. By purchasing this solution, implementers get a turn-key solution maintained by the designers themselves. However, the nature of developing a persistent virtual mirror world for simulation and training implies a long term commitment to the life of the world. Over the life of the persistent world, being locked into one software vendor can equate to an enormous expenditure of budget, which goes against the military's desire to move to open sources and open standards. It further restricts reusability and interoperability that is not necessary, and is not true about using an open standard such as .X3D.

b. Model Format

Although model translation is easily achieved between X3D and OLIVE's model format, .X3D should be a file extension natively supported to best facilitate the requirements specified in this thesis. Since the model format used by OLIVE is proprietary and integrated into the game engine, this is unlikely to occur. This means that intermediary steps and programs have to be used.

c. Terrain Format

Similar to model formatting, being able to natively translate X3D earth files, NASA World Wind, or a Google KML file into OLIVE's format is essential for the development of a persistent virtual environment that mirrors the real world. This capability does not exist. OLIVE's environment is based more upon a digitization of the real world than Second Life and There! is, but it still does not meet the requirements of a mirror world.

3. Recommendations and Uses

OLIVE has the advantage of being designed for the use this thesis envisions, as opposed to Second Life and other social networking and simulation packages being primarily designed for entertainment. These programs are then adapted to fill requirements as an afterthought. However, OLIVE's closed and proprietary format make it a useful consideration only for solutions where a dedicated system maintained by

software designers is required. Using OLIVE can provide an essential tool for simulator training in certain aspects of military and educational use, but does not meet the requirements proposed by this thesis.

C. THERE!

1. Advantages

a. Population

There! boasts a membership of 500,000 users with more joining everyday. Online usage tended to be around 2,000 members during the course of this thesis, which ranks second in user base to Second Life but ahead of the other online virtual environments analyzed in this thesis. Certain aspects, such as recruiting, would require a large user data base over all other factors.

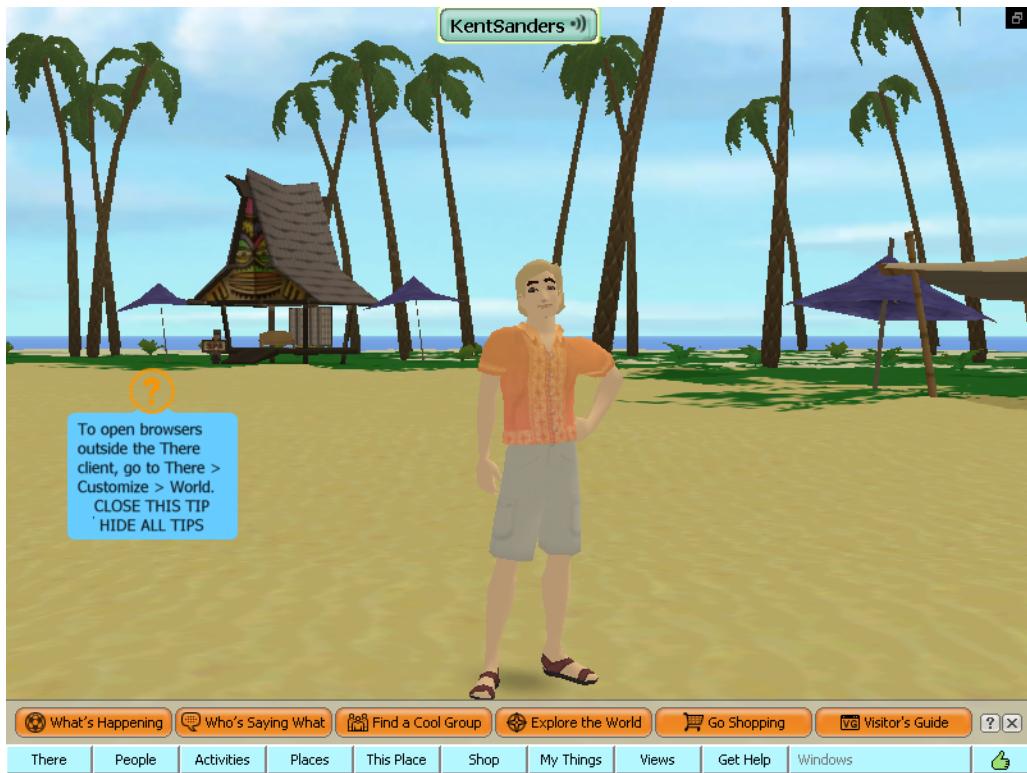


Figure 22. There are many options for avatar creation in Makena's There!

b. Recordable

There! also boasts a native means of recording both snapshots and video recording, without the use of external scripts or programs as in Second Life. This saves on computing resources and is more intuitive to use. Figure 22 shows a snapshot caption of a There! avatar standing on the beach.

c. World Creation

Where Second Life only allows users to buy their own private island for more control over simulation, There! allows users to create an entire world. This exclusivity and level of control might be required for many applications of serious gaming. Allowing social networking users to move in between the worlds of There! and separately-owned private worlds combines the best of a public world virtual environment with the aspects of a private world. An example of the There! engine powering a private world can be found at MTV’s Virtual Laguna Beach.

d. Weather Control

As important as the weather can be to the conduct of military operations, so too can weather be important in the simulation of those operations. In particular, control of the weather can be an especially critical factor. There! provides comprehensive weather and climate controls more powerful than “forcing sunrise” in Second Life. While none of the simulations analyzed in this thesis had accurate weather *modeling*, being able to force a change in weather as necessary for a module or simulation is a helpful compromise.

e. 3DS Max Support

There! does allow development of models in the large modeling software formats, with import capability. This makes model translation and importation only a few steps away, assuming the model meets Makena’s requirements on level of detail acceptability and copyright and texture requirements.

f. Voice Communications

At the time of this thesis, Second Life had no voice chat capabilities and users were relying on third party programs such as TeamSpeak and Ventrilo to accomplish voice. There! has native voice for preferred users, meaning there is a small cost to use, that is well accomplished and well defined, though not so refined as OLIVE's multi channel global voice solution.

g. Appearance



Figure 23. There! has a uniform world appearance.

By controlling all models and textures that are imported into their developed world, the game designers are able to keep a more uniform and appealing image compared to the chaotic juxtaposition that often occurs in the regions of Second

Life. Figure 23 shows the consistent graphical design that can be found throughout the There! universe without the chaotic clash of designers found in other user-created simulations.

2. Disadvantages

a. Limited Scripting

With There! being primarily a vehicle for social networking and chat, there are limited options available to create a truly useful tool for simulation and training.

b. Model Submission

All models must first be submitted to Makena Technologies to be approved prior to addition to the There! space. This is designed to prevent copyrighted content from entering the virtual world, as well as to help create a more uniform appearance. This works to some extent in that the world is more uniform than the wild juxtaposition of landscapes that can be found in Second Life, but it also results in a more tame landscape with a non-efficient two step process of creation.

c. Commercial

There! is a commercial project from design to implementation, with not a lot of regard for any use but entertainment and social networking. It is designed as a money-making vehicle by the developers to enable chat and other social contact between its participants. It is apparent that not a lot of design has been geared toward education, research, and other functional utilization.

3. Recommendations and Uses

There! is primarily a social networking solution. It is often billed as a competitor to Second Life, yet without the sophisticated scripting technique and wide open configurability of Second Life, There! feels like a 3 dimensional chat room. It has been used well to extend other social situations such as its *Laguna Beach* world, and for

commercial advertising endeavors. However, ultimately it lacks the sophistication, adaptability, and growth potential needed to fill most of the roles propositioned by this thesis.

D. SECOND LIFE

1. Advantages

a. *Population*

At the time of this thesis, Linden Labs claims nearly 8 million total residents, with an active user load of 32,000 online at a time. Its popularity continues to grow as awareness and complexity of the simulation grow. This large player base means a large number of developers are also bringing different ideas, models, and experiences to the simulation. For uses that require reaching a large audience, this makes Second Life the prime candidate.

b. *Open Source*

Linden Labs has already made available the client code for its software and has promised to release the server code as open source as well. Until that source code is released, being only able to modify the client code restricts developers to only controlling how the users interact with that world, and do not adequately give them control over how that world itself behaves.

c. *Requirements*

Second Life requirements are low and non-demanding. It is capable of being rendered on machines several years older than current architecture. Like There!, it also minimizes network traffic to be effective and viable using a dial-up telephone connection.

d. Hosted

Second Life is imagined, created, and owned by its residents, and the servers are maintained by Linden Labs. Creating a functional footprint in Second Life can be accomplished for a few dollars a month, and has the potential to reach millions of residents.

e. Web Integration

Second Life is connected with the web and in-world links open up the default web browser. The reverse is also true; clicking on a “SLurl” (Second Life URL)¹⁹ in the web browser will open Second Life and take the user to the selected point of interest. Although other simulations will typically allow for browser integration, none are as seamless or two-way as that developed by Linden Labs. For example, clicking on a SLurl while not being logged in is expected to open the Second Life client, login if needed, and take the user to the selected point of interest.

f. Movement

Movement within the virtual world is designed to keep an individual’s avatar at the current point of interest. Teleporting from point to point, flying, inviting others to teleport to the user’s location or being teleported to their location are all allowed within Second Life. This makes the large terrain of Second Life accessible to any point in the world from any point in the world.

¹⁹ SLurl, “Location Based Linking in Second Life,” <http://SLurl.com/secondlife/Green/190/153/20/>, last accessed August 2007.



Figure 24. The De-babbler script is translating local text into another language from within Second Life.

g. Tutorials

There exists an extensive set of tutorial videos about Second Life created by residents and submitted to YouTube. Videos exist for difficult tasks such as learning to build with the in-world system and using the scripting language, to the more mundane tasks such as how to manage in-game inventory and navigate the world. Tutorials are very helpful for the difficult and complex user interface of Second Life, as demonstrated in Figure 24.

h. Avatar Customization

Although the initial modifications made available at the time of avatar creation are somewhat limited compared to specialized avatar creation software, both free

and purchasable modifications can be made to an avatar after creation that are limited primarily by imagination. Like most of the other benefits of Second Life, this is both an advantage and a disadvantage in that some avatar modifications are beyond distracting (to say the least) if used as part of a serious application or simulation. Figure 25 depicts a somewhat tame avatar being created.



Figure 25. Second Life has the most configurable avatar of the popular simulations.

2. Disadvantages

a. Graphically Unappealing

The same low computing and graphical requirements that allow the simulation to be easily rendered also count against its appeal. Models must maintain a low polygon count, resulting in objects being primarily composed of geometric

primitives. Texture resolution and colors are also lower resolution, and without vertices graphical features like pixel shading can't be accomplished.

b. Chaotic Content

A world that is built piecemeal by its 8 million residents like Second Life can lead to jarring juxtapositions of simulations within close proximity. Education simulations might be surrounded by gambling, nightclubs, and other adult themed simulations. The only true means of controlling an individual's neighborhood is through purchasing a private island. This can reduce the objectionable content that a designer is attempting to eliminate, but does not prevent an offensive avatar from wandering through the public simulation. This results in frequent and severe interruptions that often make purposeful communication difficult in any well trafficked area. This becomes a trade-off of allowing everyone access but being forced to endure the "creativity" of some users, to stifling that creativity to develop a more traditional real-world environment.

c. Voice

Only with the latest client release has voice been made available to residents in Second Life. Beta testing the Voice over Internet Protocol (VOIP) was problematic and often was simply not available. When it is operating, the voice rendering is spatially aware to recreate the effect of speaking at distances, as well as a "group broadcast" mode to allow communication with everyone in the user group regardless of geographical location. Previously, the voice programs TeamSpeak, Ventrilo, or Skype had to be separately configured and used to allow this functionality.

d. Security

As with any large gathering of diverse people, there are conflicts. Second Life is no exception. With a weekly crime blotter that involves such attacks as avatar murder, particle bombing, and even avatar rape, Second Life can sometimes be a study in the ugliest aspects of virtual reality. Virtual boundaries established to prevent avatars from entering in to private areas have also been overcome. Security of Second Life lags

behind security of the World Wide Web, but is improving. Still, it would be extremely unwise to bring any sensitive data into Second Life.

e. Terrain

Buying already-created geography means users are confined to the existing terrain, with only some minor alterations allowed. There are also only four basic models of private islands to choose from, although Linden Labs does have an option to create a new island based on a provided terrain file. There is no provision for matching the simulation engine with real-world terrain provided by Google Earth, NASA World Wind, or X3D Earth.

f. No Playback

With no internal means of recording or replaying the simulation, training and education benefits are limited. Solutions that have come up to mimic replaying include using external software to capture and create a movie of the simulation, or to simply loop the simulation again and again.

g. Proprietary

Linden Labs continues to use proprietary model architecture to provide a world that transmits well over dial up connectivity. However, this proprietary system makes using existing model databases, using better designed modeling software, and inter-connectivity between virtual environments an impossibility.

h. Cost

Land prices inside Second Life must be considered when considering the primitive count allowed is based on virtual square feet. To establish a long-term virtual presence of appropriate size and availability for military or educational use inside of Second Life, a private island is required. The initial purchase of such an island costs over a thousand dollars, with a monthly fee for the island to be maintained. While this fee is

often feasible, and is much lower than a commercial simulation will cost, it is still not a free solution, thus scalability is greatly inhibited.

3. Recommendations and Uses

a. Language Training

If the best way to learn a foreign language is to be immersed in it, as Johnson claims in *Serious Games for Language Learning: How Much Game, How Much AI?*, then Second Life is poised to become a potent tool in language education. By creating authentic geography, models, attire, and avatars from a specific region, users can “immerse” themselves in a foreign culture without ever leaving their home.

Powerful tools such as IBM’s Linguaphile²⁰ or the in-world script “Babbler” and “Debabbler” provide language encoding and decoding in real time to facilitate communication using text messaging. Babbler and De-babbler send in game text to Google Translate via a web bridge, and then return the answer as translated chat. This is a great example of the seamless integration between Second Life and the World Wide Web, and how harnessing existing tools of the two dimensional web can be accomplished.

b. Distance Education

Educators, second only to commercial developers, have been quick to embrace the possibilities of Second Life. Entire campuses have been recreated and classes held virtually within Second Life are typically free to the user. With libraries and campuses moving to virtual space to save money on real estate, online education is expected to continue to evolve and flourish.

This is also useful for military members often stationed away from college campuses, but still connected to the internet. Within Second Life there exist programs to

²⁰ IBM, “Adding Simple Translation to Second Life,” <http://www.ibm.com/developerworks/library/l-second-life-3.html#N10061>, last accessed July 2007.

prepare for examinations to earn a real world general equivalency diploma (GED), to apply for U.S. citizenship, or to learn English as a second language.²¹

c. Recruiting

The possibility of educating potential recruits on job possibilities and requirements, while still maintaining an entertaining environment to attract interested individuals make Second Life an appealing choice for recruiting. Similar to the America's Army game project, entertainment and gaming attract large numbers that are primarily members of the recruiting demographic. Creating a military complex in-world that allows residents to test drive vehicles, experience simulated military jobs, and other tasks can be accomplished with little time and investment. These locations will then link back to the existing web sites through SLurls, as well as have their locations linked via HTML.

d. Teleconferencing / Virtual Collaboration

Many in-world examples of teleconferencing and virtual collaboration have been documented and made available via YouTube. Virtual concerts have been held, meetings and summits taken place, and entire college courses have been virtually held within the world of Second Life.

E. PROJECT DARKSTAR

1. Advantages

a. Open Source Java

One of the key aspects of using a solution like Project Darkstar is having access to the source code. No solution developed by another company for their own and adapted for military use is going to contain every feature and operation that will be

²¹SimTeach, "Second Life Education Wiki," http://www.simteach.com/wiki/index.php?title=Second_Life_Education_Wiki, last accessed August 2007.

necessary for military simulation, as envisioned by that user. The options then become contracting a company to create this application to the end user's specifications, which again locks purchasers into long-term, expensive contracts, or to take the existing project and tailor it to the end user's individual needs and make do with a mismatch of requirements and features. As development costs are driven down by projects such as Project Darkstar, creating a better-fitting virtual environment that meets all objectives and requirements becomes a more appealing choice.

b. Developing User Community

As is the strength with all open source software, the community that uses and extends its functionality provides free resources and innovation that will not be developed by other means. Sun System's goal of developing virtual office environments in its Project Wonderland demonstration (the software layers are depicted in Figure 26), fits well with some of the applicable uses for military and educational institutions. These uses, as well as the many other uses that can be expected to develop from the user community, come at very little cost to developers.

World	MPK20: Sun's Virtual Workplace <ul style="list-style-type: none"> • World customized to support Sun's distributed workforce • Includes applications for sharing and collaboration 	
Client	Project Wonderland <ul style="list-style-type: none"> • Open source Java 3D-based graphics engine • Manages world, animation, and avatars • Supports application sharing (initially Java and X applications) • Extensible and customizable worlds 	Software Phone
Server	Project Darkstar <ul style="list-style-type: none"> • Communication and application framework • Targeted at games • Highly scalable • Handles persistence • Allows extensible set of core services 	Voice Bridge

Figure 26. MPK20 provides a great example of Project Darkstar's capabilities [From Sun System's *Project Darkstar*].

c. Configurability

No other simulation analyzed in this thesis provides the level of configurability and customization of the Project Darkstar environment. The software has been developed from its inception as a fully configurable tool used to deploy other MMOG games without requiring the game developer to “reinvent the wheel” of basic MMOG architecture. When Second Life releases the source code for its server architecture, it will still come with its core design components embedded in the software, making it a much larger project for software developers to implement their own flavor of solutions. Most of the other simulations and applications have closed source systems, allowing for no customization or configuration at all.

d. Cost

Project Darkstar is open source and released under the GNU version 2.0 license. The Project Darkstar community is also well established to freely share third-party developed solutions amongst community members since there does not exist an in-game economy to profit from (as in Second Life and There!), nor is the environment proprietary (as in OLIVE). Figure 27 shows the freely distributed MPK20 world using the Project Wonderland client and Project Darkstar server structure.

e. Terrain

To develop a mirror world, one must have complete control of the terrain in order to create and update a digitization of the real world. Second Life and There! have created virtual terrain that does not reflect the real world, and could not be used for this application. OLIVE has created a virtualized environment in a mockup of the real world, but is not a true mirror world. Since OLIVE is a propriety solution, it could not be modified by developers to fulfill this requirement without support and cost. By having the complete control over the environment that Project Darkstar allows, it becomes possible to develop a fully persistent mirror world.



Figure 27. MPK20 is a virtual building extending upon IBM's Menlo Park facility using Project Darkstar.

f. Plug-in Design

With a solution such as Second Life, There!, and OLIVE, developers inherit the existing game engine, AI, physics, and other aspects of the virtual world. While reducing cost and development time, this offers less flexibility, modifiability, and configuration that might eventually become prohibitive. Having control over these crucial elements of the simulation allow not only for control over development, but an easy path to upgrading should a free and better solution be developed by a third party.

g. Scalability

Project Darkstar is written for a game scalability that is unprecedented in other software projects, including MMOGs. Recognizing that scalability has killed many

good game projects, Sun has anticipated player growth beyond what systems are currently designed for. This scalability is essential for the development of a simulation that meets the thesis requirements.

h. Persistence

Project Darkstar is inherently persistent in all game objects, which leaves the game developer free to not worry about database management.

i. Fault Tolerance

Sun has programmed built-in fault tolerance to the game engine, so if a server fails the game continues to function and is transparent to the user. For the purposes of a serious game or military simulation, this level of reliability and error correction is essential.

2. Disadvantages

a. Work In Progress

Although Project Darkstar provides a strong basis for a launch point, it is still in beta and is an unfinished and untested work. Developing an application using this architecture will necessitate developing most features in house. As with many of the disadvantages provided for this simulation comparison, this can also be a distinct advantage for the software developers, depending upon the application they are hoping to develop.

b. No User Scripting

Having control over the source code would allow game developers or third party developers to develop an in-game scripting interface for generating models or scripts, but currently this does not exist. This severely limits the ability for users to create their own tools in-world, as is often found in Second Life.

c. Limited Documentation

The Project Darkstar documentation, the forum, and the user community continues to grow daily, but currently the best means to learn to use the tools Project Darkstar provides is by looking directly at the source code. Looking directly at source code to find answers to development questions typically prevents average users and developers from contributing to the project.

3. Recommendations and Uses

a. Persistent Mirror World Simulation

Despite being an unpolished work in development, there is much about Project Darkstar and Wonderland that will lend itself to being the best choice to answer the questions developed by this thesis. Darkstar has native X3D abilities, and can pass and recognize XML and DIS packets. It provides a strong basis for persistence and networking, and the developing shared office applications point to very promising applications. It is open source and allows for open standards providing a near-limitless amount of control and development. It is the only simulator studied in this thesis that had the potential to match all the requirements propositioned. It is difficult to estimate the cost of development in each of the simulations, but due to the open source nature of the software, it could be expected that Project Darkstar would also be the cheapest solution to the questions posed by this thesis. It stands out as the clear choice.

F. CONCLUSION

During the course of this thesis examination, many social networking and simulation programs are analyzed. This chapter makes informal comparisons between these simulations and existing social networking tools to see if the programs features match the requirements set forth by this thesis. Key advantages and disadvantages, as well as possible military and educational use are provided for each tool.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

From the many social network applications, games, and simulation programs available today, and ever more being developed as the field of serious gaming and simulation expands, there exist some clear advantages and disadvantages with each of these programs. None of them fully answer the needs, flexibility, and versatility of a fully persistent mirror world virtual environment, as seen in Table 1. Each of the analyzed simulations might play a piece in the total virtual environment / simulation of the future military, but none provide the extensibility and scalability to answer the questions analyzed in this thesis without development.

However, it is possible using the open source standards and software available, to develop a long term solution in the form of a persistent mirror world virtual environment to answer the specific and growing needs of the military simulation and virtual environment community. The clear solution to this development is Project Darkstar. Its open source nature, flexible design, MMOG architecture, and growing user community make it an ideal choice to fulfill the requirements set forth by this thesis. By maintaining these projects within open source standards, the virtual environment and simulation community can anticipate the coming Metaverse of networked simulations and social networks to maximize its effort in each of its individual simulations, and capitalize on features developed in other applications.

Matching the open standards of X3D with Project Darkstar using a mirror world environment provided by programs such as X3D Earth or Google Earth creates a tool with incredible potential. The functionality and usefulness of a fully persistent 3D world that is owned and developed by the military, for military use and connected to other military simulations provides increasing returns over the functionality of each of those stand-alone simulations.

SIMULATION					
Features		OLIVE	Project DarkStar	Second Life	There
	DIS/HLA Connectivity	X	X		
	Free Development		X		
	Free Usage		X	X	X
	In-world Model Creation			X	
	In-world Script Creation			X	
	Massively Multiplayer	X	X	X	X
	Modifiable Terrain		X	X (limited)	X (limited)
	Native VOIP	X	X		X
	Offline Functionality		X		
	Open Source		X	X (client)	
	Recordable Playback				X
	Weather Control				X
	X3D Format		X		

Table 1. Key features that are available in each simulation are denoted by an “X”

B. RECOMMENDATIONS / FOLLOW ON WORK

1. Develop a Fully Persistent Mirror World Simulator

Taking the lessons learned and knowledge acquired from this thesis, follow on work to create a fully persistent, three dimensional simulation with the entire real world as the virtual environment can be accomplished. The following is a list of some of the basic features and issues to be considered when commencing such a project.

a. *Where to Start?*

Project Darkstar has created the groundwork for a versatile and capable system of client / server architecture, but it is not complete. The project was launched as a way to provide the basis for smaller companies to inherit the architecture of a MMOG and develop the workings of the game within this environment. This includes persistent object handling, which is not easily done, and is inherent in Project Darkstar. It includes rudimentary physics, networking, a GUI, and artificial intelligence, but leaves these areas open for development for each game or simulation as necessary. These aspects that are left not well-defined are aspects that developers would find restrictive if they *were* well-defined.

Sun also claims that Project Darkstar can support a server farm to handle the user load of a MMOG; however, documentation and actual implementation do not seem to be in place at the time of this writing, and would make for a more difficult installation than if these documents were in place. Sun is also working on developing a turnkey solution to their Darkstar environment, offering to develop the server hardware that accompanies the software. It is to this end that they have open sourced the Project Darkstar code.

Installed teleconferencing applications, and other Project Wonderland / MPK20 demonstration features, did not seem fully functional at the time of this thesis, even when run on the appropriate version of Gentoo Linux. Windows support for the client was even less robust.

The software is open source and written primarily in Java, with some C++, so it does provide the best starting point of all the software methods studied. Perhaps this advantage will be less prominent if Second Life open sources their server code, or another game engine such as the Torque Game Engine begin to shift focus towards a more serious game-oriented position, but it currently makes Project Darkstar the clear favorite.

b. Networking

The foremost difficulty of developing a mirror world environment as propositioned in this thesis is overcoming network bandwidth issues. As discussed earlier in the Metaverse Roadmap article, these constraints will decrease in the future as broadband access and other high speed connections become more ubiquitous, but will always likely be a factor. Amazing things are being done to bypass these constraints (for example Second Life's minimal network traffic). For a very in depth review of this network bandwidth problem, which is beyond the scope of this thesis, *A Taxonomy of Networked Virtual Environments* by Michael Macedonia and Michael Zyda give enormous insight into the difficulties and possibilities of networked virtual environment for military training applications.

To run the software, especially for such a diverse world as this thesis proposes, developers will find MMOG architecture invaluable. This will require login server architecture with appropriate switching, and servers to control appropriate portions of the mirror world. This will be trickier than normal MMOG architecture which uses zones of roughly approximate size to handle a well controlled player base. In this system of virtual world for training, this will work for the majority of time when users were running their own lessons or developing in their own niches within the environment. However, when a large scale training or simulation was to be run, the majority of the users will want to flood to one location and hence one server. This will rapidly overwhelm any individual server based upon MMOG architecture. Project Darkstar is the only simulation evaluated in this thesis with the massive scalability necessary to tackle issues such as these.

The goal to fully represent the real world in a mirror world environment, including largely uninhabited and unutilized places such as the Arctic and Antarctic regions, is truly a problem of scale. In order to save server resources, these largely uninhabited regions will need to be larger than normal server regions and approximated through clever algorithms, as busy locations such as Norfolk will need to be on a much smaller server region (for server load) and in much higher detail. With limited documentation provided by Sun that directly references their server architecture, it is unsure that their engine currently has these measures in place since most MMOG architecture is designed for server load balance, including terrain size. However, with Sun's work being open source, such modifications can be made at a later date when the user load does become an issue, well into the deployment cycle of the software, and provides a means to springboard the development of the virtual world when the user base is very small.

This virtual world will also need to accommodate all forms of internet connectivity as well as provide some means for offline capability to be truly effective. This will be a difficult constraint to achieve and will require always implementing the latest compression algorithms for online connectivity to ensure minimal network traffic. While online, the virtual world must be accessible to remote units that are using often slow and unstable satellite connectivity but yet remain synchronized within the simulation to faster, better connected units on high bandwidth land connections. In order to make itself available to the front line units, there must exist a way to run portions of the world and simulations while offline completely. Project Darkstar is the only simulation studied that allowed local server operation in the event that offline connectivity is required.

c. Environment

Another key difference between a virtual mirror world simulation and a MMOG game is the fluid state of the terrain of the zones. In a MMOG, the zones will remain largely unchanged and will be taken offline for maintenance and environment changes. In the mirror world simulation, developers want the zone to be able to be

changed with any real world changes, without having to be taken offline. The changes will have to come from not just the world maintainers and developers, but also users with appropriate access. This will necessitate some form of connection with a world modeling project like X3D, ideally built within the world to allow for in-world content creation and modification. It will also be necessary to be able to convert and import files and geometry from the other major earth modeling programs such as Google Earth, NASA World Wind, and Microsoft Virtual Earth. These changes will also then need to be saved to the server and updated with every user in that zone. A good example of how to program the world updating might become available when Second Life server code is open sourced. For an existing example of how this interoperability extends the functionality over any single program, Planet 9's Virtual Earth 3.0 platform interacting with Google Earth, X3D Earth, Microsoft Virtual Earth, or NASA World Wind is an incredible example of the power of this newly developing tool.

Model importing and exporting is also a key feature to develop. Since Project Darkstar can natively import X3D models and other formats, it provides an excellent means to harness the existing libraries of SAVAGE and other military models already developed. It also provides a model translation path to all of the other major 3d file formats such as those used by 3DS Max and Maya. This is an advantage not present in any of the other environments and simulations studied in this thesis without using another software converter or loading application.

Ultimately, some form of automatic updating to the world via unmanned vehicles will need to be developed. Although Google has experimented with automatically uploading world content on manned vehicles with cameras, the true power and future of automatic world generation will harness the growing fleet of unmanned vehicle sensors on land, in the air, and on the sea. It is difficult to predict what in what direction these technologies will come, beyond the directions provided by the Metaverse Roadmap, but there can exist no doubt that this functionality will become a cornerstone of the mirror world development.

2. Recommended Uses

As with any virtual environment, the list of possible uses is far too long to be exhaustively listed. However, the following are some of the essential uses that a mirror world virtual environment can fill:

a. War Gaming

This system will allow for war gaming and training at an unprecedented level. No longer will war gaming be an advanced tool learned late in a military career, but rather a normal part of junior officer training conducted at each of the service academies, the Naval War College, and Naval Postgraduate School. Key battles, tactics, and strategies that are assigned for military officer study could be simulated to allow the officer multiple chances to make those tactical and strategic decisions themselves.

b. Distance Learning

The future of the classroom might be virtual environments. Every year more and more courses are moved to the online realm, and in often cases entire courses and sometimes schools have become online-only resources. The virtual environment is a natural extension of that trend, incorporating already existing resources such as forums, whiteboards, blackboards, html, video, and other 2D resources. It presents this material in a higher level of interoperability and interaction than existing methods. Second Life is a key example of the educational benefits of such a learning system.

Furthermore, some courses and material due to its lethal or costly nature, is best suited for training in a virtual environment. By driving down the cost of such training, allowing for mistakes to be made that do not result in death, and a form of replay of the training held, virtual environments will rapidly replace some existing forms of training. Furthermore, serious games can be developed that actually involve the user to gain the benefits of training previously conducted in this thesis. The military uses countless hours of non engaging short film training (General Military Training or GMT) that discusses everything from proper grooming habits, sexual harassment sensitivity,

information security, and health issues related to sexually transmitted diseases. Often these films are not involving, and are played on site TV with the assumption the military member will assimilate it by having it playing in the background. How much more effective could this training be if it were conducted as a required game session each member sits through? Sexual harassment training when “gamed” will become far more involved than when merely viewed. Once these systems are developed, reusability and extensibility come at no additional cost. Networking and connecting all forms of simulation and learning reusing existing tools provides increased functionality at little development cost.

c. Recruitment

Elements of the simulation could exist with a duality of purpose, much like America’s Army. According to Michael, portions of America’s Army are used primarily as a recruiting tool, while other aspects of the simulation are very effective squad leader training tools. A virtual environment mirror world will also bring all of these aspects to the casual user, potential recruit, and senior decision maker, with content, security, and accessibility tailored to each. As no two users use Second Life in the same fashion, no two users will use a serious game simulation in the same fashion.

d. Social Networking

The arena of social networking, much like other key factors studied by this thesis, is growing at an exponential rate. However, much like the simulations and virtual environments, these social networks have needlessly created a “walled garden,” restricting connectivity to other social networking systems. Geist points out in *Pull Down the Walled Gardens* that MySpace does not freely interact with Friendster or Facebook, yet all of these sites are web-based systems which means achieving interoperability was an essential element that had been inherited with web architecture but consciously eliminated. Military members that are often deployed have shown a propensity to using these systems to maintain contact with their geographically separated loved ones. Recently, this type of contact has been restricted due to issues of bandwidth

and security. However, where security is the only consideration, this virtual world can be used to help bridge the gap between deployed soldiers and sailors and loved ones at home in an environment that is better controlled than the freely available social networks listed above. As an example, a deployed soldier could login to the simulation while in his off duty hours and have his own 3D environment virtualized, and connected to the mirror world, where he can meet his significant other in avatar format. They can chat and socialize, play games, and interact with each other much like Second Life avatars. That same virtual environment would also contain learning objectives, areas of interest, and possibly even elements from his military job in one location.

One reason to create a separate world is that, with the exception of OLIVE, all the other virtual world simulations are still primed toward being fun. However, sometimes fun must take a back seat to realism and simulation accuracy for military and other serious use. The reverse is also true in that educators have to enliven some of training to make it more palatable. Despite some similarities, serious games are different from entertainment games, and need to be approached with these differences in mind.

e. Information

The existing simulations being developed and analyzed within this thesis are all one directional interfaces of “fishbowl virtual reality”; however, by analyzing developing systems of Future Force Warrior for instance, it can be seen that the push in the military is to redistribute this information back to the war fighter. For an example, the Second Earth project features a Second Life simulation based on real world geography of Google Earth, but it is not designed to project that information back to the user in a system that could be useful for a head mounted display for navigation, operation, and other in-field uses. This will be one of the advantages of a system designed for both online and offline use. This can be accomplished on devices as complex as heads up displays, head mounted augmented reality systems, or on something as simple as a GPS enabled cell phone, similar to Planet 9’s Raygun.²²

²² Planet 9 Studios, Raygun, <http://raygun.planet9.com>, last accessed August 2007.

Most military personnel are not involved in frontline combat. The support roles include intelligence gatherers, cooks, supply and logistics personnel, etc. who are doing their jobs under extremely adverse conditions. These support personnel have been killed at a higher rate than the combat personnel in the Iraq war. Serious games can be utilized to train these support personnel as well as the combatant, which is an area that is largely unaddressed.

f. Military Officer Training

The existing simulations being developed and analyzed within this thesis are all one directional interfaces of “fishbowl virtual reality”; however, by analyzing developing systems of Future Force Warrior for instance, it can be seen that the push in the military is to redistribute this information back to the war fighter. For an example,

3. Establish a Second Life Recruiting and Campus Presence

a. Campus Presence

An area that educational institutions can benefit from developing a virtual environment presence is becoming involved in the Second Life campus projects. Many campuses are beginning to explore this new educational tool to best determine how it can be used. As with most educationally developed projects, these tools are made freely available to other educators. The military can also benefit greatly from developing a presence inside Second Life, which places the MOVES institute in a strong position to develop this resource, as it uniquely straddles the educational and military communities within one campus.

There are many interesting questions that can be asked about presence, training, immersion, and other questions about operating within a virtual environment. Second Life provides one of the best means of answering these simulation research questions by removing the development of a virtual environment from the setup of the study.

Due to the nature of the military lifestyle, military members often must use a form of distance education to advance their education while being forward deployed to remote locations away from college campuses. This places the military member at the forefront of distance learners, and educators hope to use Second Life to provide an array of distance learning tools.

b. Recruiting

Just as a webpage is an extremely cost effective means of providing information to a wide user base, so too can a free simulation engine like Second Life be harnessed. This has the potential to make it a very important recruiting tool to better show interested parties what military life is like. Developing a military presence inside Second Life that is linked externally to existing recruiting websites can provide one of the most cost effective means of recruiting in an era where the military is very interested in ensuring appropriate manning levels are maintained but on cost conscious recruiting budgets.

c. Public Relations

Creating a military presence within Second Life can be a positive Public Relations experience. Providing a means to virtually tour the USS Arizona memorial in Pearl Harbor to users who might never view the real memorial, all at no cost to the user, is an excellent means to pass on military heritage and tradition to not just the casual observer, but the potential recruit as well.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Brutzman, D. and Daly, L. (2007) X3D for Web Authors. Morgan Kaufmann Publishers.
- Brutzman, D (1997) Graphics Internetworking: Bottlenecks and Breakthroughs. In Clark Dodsworth Digital Illusions p. 61-97.
- Capps, Michael, Perry McDowell and Michael Zyda (2001) A Future for Entertainment – Defense Research Collaboration. IEEE Computer Graphics and Applications p. 37 - 43.
- Geist, M., Pull Down the Walled Gardens (2007, August 15) online BBC News.
<http://news.bbc.co.uk/2/hi/technology/6944653.stm/>, last accessed August 2007.
- Johnson, W. L., Vilhjalmsson, H., and Marsella, S., (2005) Serious Games for Language Learning: How Much Game, How Much AI? In *Artificial Intelligence in Education*.
- Koster, R., (2004) Theory of Fun for Game Design (ILLUSTRATED). Paraglyph Publishing.
- Macedonia, M.R., et al (1994) NPSNET: A Network Software Architecture for Large Scale Virtual Environments/. In *Presence* Vol. 3, No. 4.
- Macedonia, M. R., and Zyda, M., (1997) A Taxonomy for Networked Virtual Environments. In *IEEE Multimedia* Vol. 4, Issue 1 p. 48 – 56.
- McGregor, D., Kapolka, A., Zyda, M., Brutzman, D., (2003) Requirements for Large-Scale Networked Virtual Environments. In *ConTel 2003, Proceedings of the 7th International Conference on Telecommunications* Vol. 1 353 – 358.
- Mennecke, B., et al (2007) Second Life and other Virtual Worlds: A Roadmap for Research from 28th International Conference on Information Systems (ICIS)
<http://ssrn.com/abstract=1021441>, last accessed December 2007.
- Michael, D. R., (2005) Serious Games: Games That Educate, Train, and Inform Course Technology PTR.
- Nicklaus, S., (September, 2001) Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE) Naval Postgraduate School, Monterey, CA.
- Prensky, M., (2004) Digital Game-Based Learning. McGraw-Hill
- Prensky, M., (October, 2001) Digital Natives, Digital Immigrants. In *On the Horizon* Vol. 9, No. 5 p. 1 – 5.

Rauch, Travis Savage Modeling and Analysis Language (SMAL): Metadata for Tactical Simulations and X3D Visualizations Naval Postgraduate School, Monterey, CA, March 2006.

Rymaszewski, M., et al. (2007) Second Life: The Official Guide. John Wiley and Sons.

Smart, J., et al (2007) Metaverse Roadmap: Pathways to the 3D Web. A Cross-Industry Public Foresight Project Metaverse Roadmap
<http://www.metaverseroadmap.org/overview>, last accessed August 2007.

Smith, R., (2006) Game Impact Theory: The Five Forces That Are Driving the Adoption of Game Technologies Within Multiple Established Industries. CTO Net
http://www.ctonet.org/documents/SmithR_GameImpactTheory.pdf.

Stitt, J., and Chappell, L., (January, 2005) Games That Make Leaders: Top Researchers on the Rise of Play in Business and Education. Wisconsin Technology Network
<http://wistechology.com/article.php?id=1504>, last accessed August 2007.

Terdiman, D., (December, 2005) New Gateway to Multiple Virtual Worlds? CNet News
http://news.com.com/New+gateway+to+multiple+virtual+worlds/2100-1043_3-5983264.html, last accessed August 2007.

Thorpe, J. A., (1987) The New Technology of Large Scale Simulator Networking:
Implications for Mastering the Art of Warfighting In *Simulation Conference Proceedings* p. 770 – 774.

Zyda, M., et al (2003) Entertainment R&D for Defense in *IEEE Computer Graphics and Applications*. P. 28 – 36.

Zyda, M., and Brutzman, D., (1997) *At Least Ten Unsolved Problems in Networking Large-Scale Virtual Environments (LSVEs)* presented at Tele-Immersion Workshop <http://gamepipe.usc.edu/~zyda/presentations/LSVETen.pdf>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Don Brutzman
Naval Postgraduate School
Monterey, California
4. Amela Sadagic
Naval Postgraduate School
Monterey, California
5. Terry Norbraten
Naval Postgraduate School
Monterey, California
6. Rita Turkowski
Web3D Consortium
San Francisco, California
7. David Collen
Planet 9 Studios
San Francisco, California
8. Yvonne Masakowski
Naval Undersea Warfare Center
Newport, Rhode Island
9. James Oliverio
University of Florida
Gainesville, Florida